
Disentangling Interactions and Dependencies in Feature Attribution

Gunnar König*
University of Tübingen
Tübingen AI Center

Eric Günther*
University of Tübingen
Tübingen AI Center

Ulrike von Luxburg
University of Tübingen
Tübingen AI Center

Abstract

In explainable machine learning, global feature importance methods try to determine how much each individual feature contributes to predicting the target variable, resulting in one importance score for each feature. But often, predicting the target variable requires interactions between several features (such as in the XOR function), and features might have complex statistical dependencies that allow to partially replace one feature with another one. In commonly used feature importance scores these cooperative effects are conflated with the features’ individual contributions, making them prone to misinterpretations. In this work, we derive DIP, a new mathematical decomposition of individual feature importance scores that disentangles three components: the standalone contribution and the contributions stemming from interactions and dependencies. We show how the decomposition can be estimated in practice and propose a new visualization of feature importance scores that clearly illustrates the different contributions.

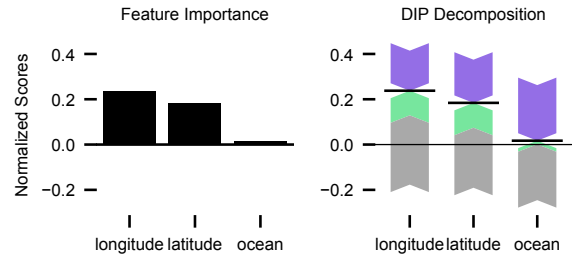


Figure 1: **Feature importance, old vs. new.** Consider a model that predicts house prices using the features *longitude*, *latitude*, and *ocean proximity*. *Left:* Leave-One-Covariate-Out scores (LOCO). *Right:* Our decomposition of the same scores (black lines) into each feature’s standalone contribution (gray) and the contributions of interactions (green) and dependencies (purple). The arrows of the bars indicate whether the contribution is positive or negative; their values sum up to the LOCO scores.

1 INTRODUCTION

Tools from explainable AI (xAI) are increasingly employed not only to explain a machine learning (ML) model’s mechanism but also to gain insight into the data generating process (DGP) (Freiesleben et al., 2024). In this context, global loss-based feature importance techniques are often used to learn about the features’ predictive power, that is, their ability to accurately predict the underlying target. To enable such

insight, the methods remove features from the model, for example, by marginalizing out variables or refitting the model, and quantify the global effect on the empirical risk (Covert et al., 2021; Molnar, 2022). This contrasts methods like SHAP (Lundberg and Lee, 2017), which explain how a specific model arrives at its prediction for a particular observation.

Existing feature importance methods try to explain the features’ joint predictive power with just one individual score for each feature. This is problematic since, commonly, the predictive power is not simply the sum of the features’ standalone contributions but also the result of cooperative forces: Interactions between several variables might unlock additional predictive power, and variable dependencies might render the different standalone contributions redundant. As such, when attributing the predictive power with just one score per feature, the individual scores conflate standalone and cooperative contributions, making them prone to misinterpretation.

The main contribution of this paper is to derive a new

*Equal contribution. Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. Copyright 2025 by the author(s).

mathematical decomposition of individual feature importance scores that disentangles the contributions of individual features and cooperative effects stemming from interactions and dependencies.

Let us consider an illustrative example. Suppose the goal is to predict the price of a house based on its longitude, latitude, and proximity to the ocean. Inspecting a typical feature importance plot such as the LOCO scores (Lei et al., 2018) (Figure 1, left) the relevance of feature cooperation does not become clear. Using our method (Figure 1, right), we can see that the features longitude and latitude have limited use alone but are highly predictive if combined with the remaining features via an *interaction* (since together they determine the exact location). We can also see that the feature ocean proximity is useful alone, but its predictive power can be replaced due to its *dependence* with the remaining variables (longitude and latitude can replace ocean proximity). For details on the implementation we refer to Appendix B.2.

In Section 4, we show that the impact of interactions and dependencies are entangled in the predictive power. To disentangle them, we first separate pure interactions and main effects *in the ML model* (Section 5). Knowing how to decompose a prediction model, we can decompose the *predictive power* of two groups of features as the sum of their respective standalone contributions and the contributions of between-group cooperation via interactions and dependencies (Section 6). The decomposition can explain the outputs of popular feature importance methods such as LOCO or SAGE (Lei et al., 2018; Covert et al., 2020), as shown in Section 7. We demonstrate its utility on real-world data in Section 8.

In contrast to existing approaches that explain the relevance of interactions for a prediction model (Lundberg et al., 2018; Sundararajan et al., 2020; Bordt and von Luxburg, 2023; Herbringer et al., 2023), we focus on learning about the relationships in the data. Thus, we explain the relevance of interactions for the predictive power instead of a specific model’s mechanism. Furthermore, we avoid marginal sampling – crucial to enable insight into the data (Chen et al., 2020; Hooker et al., 2021; Freiesleben et al., 2024). And thirdly, we explain the cooperative contributions of *both* interactions *and* dependencies.

Contributions

- We propose DIP (disentangling interactions and dependencies), a unique decomposition of \mathcal{L}^2 -loss based predictive power that explains the contributions of both interactions and dependencies between two groups, as well as new plots that clearly visualize their respective contributions (Section 6).

- In Section 4, we show that predictive power conflates interactions and dependencies. To disentangle their contributions, we show how to uniquely separate main effects and interactions *in an ML model* in Section 5. Therefore we prove that pure interactions are unique, show how they can be estimated, and prove that the main effects are unique under mild assumptions.
- In Section 7, we show that the decomposition can be used to explain popular feature importance techniques and demonstrate the method’s practical usefulness on real-world data in Section 8. A `python` implementation of the method is publicly available on `pypi` and on GitHub via <https://github.com/gcskoenig/dipd>.

2 RELATED WORK

Explanation Techniques that Attribute Interactions There is a large amount of literature on explainable AI, we refer to Molnar (2022) for an overview. Here, we focus on techniques that attribute interactions. Shapley interaction values (Grabisch and Roubens, 1999; Lundberg et al., 2018) and higher order variants of them (Sundararajan et al., 2020; Zhang et al., 2021; Herren and Hahn, 2022; Bordt and von Luxburg, 2023; Hiabu et al., 2023) are local attribution methods that attribute interactions. A range of methods for their estimation has been proposed (Fumagalli et al., 2024a,b; Muschalik et al., 2024). Local alternatives include Integrated Hessians (Janizek et al., 2021) and directional interactions (Masoomi et al., 2022); Friedman’s H-statistic (Friedman and Popescu, 2008) or GADGET (Herbringer et al., 2023) are global alternatives. In contrast to existing methods, we avoid marginal sampling techniques, which do not allow insight into the data (Hooker et al., 2021; Freiesleben et al., 2024). Furthermore, we decompose the predictive power instead of explaining a model’s predictions and disentangle the contributions of interactions and dependencies.

Functional Decomposition The generalized functional ANOVA (generalized fANOVA) decomposition (Hooker, 2007) decomposes a function into components of different interaction order. Its computation is generally hard (Li and Rabitz, 2012; Lengerich et al., 2020). The decomposition lays the foundations for generalized Sobol indices (Chastaing et al., 2015; Gao et al., 2023). While the aforementioned methods attribute every possible subset of features, we focus on explaining the cooperation between two groups of features. Our decomposition is comparatively easy to estimate and interpretable. Also, we attribute both interactions and dependencies.

Partial Information Decomposition When replacing the \mathcal{L}^2 -loss with the cross-entropy-loss, the cooperative impact becomes the interaction information (Covert et al., 2020, Appendix C). The problem of decomposing the interaction information into a redundancy and a synergy component is called partial information decomposition and is discussed in Williams and Beer (2010); Barrett (2015); Griffith and Ho (2015); Kolchinsky (2022).

Communality Analysis In a similar vein, a range of work has focused on decomposing the explained variance of linear regression models into the standalone and shared contributions of the features (Seibold and McPhee, 1979; Nathans et al., 2012; Ray-Mukherjee et al., 2014). We generalize existing results to ML models, allowing nonlinearities and interactions.

3 BACKGROUND

3.1 Notation

Throughout the paper, we consider $(X, Y) \sim P$ to be our data generating process (DGP), consisting of two random variables: the features $X = (X_1, \dots, X_d)$ in \mathbb{R}^d and the labels Y in \mathbb{R} . They are sampled from some probability measure P on $\mathbb{R}^d \times \mathbb{R}$. We assume X_1, \dots, X_d, Y as well as every prediction function to be \mathcal{L}^2 -measurable with respect to P . We denote the set of all features by $D := \{1, \dots, d\}$ and its power set by $\mathcal{P}(D)$. For a set of features $J \subseteq D$, the term \bar{J} refers to the set $D \setminus J$. For sets of just one feature, we tend to drop the brackets for readability, for example, \bar{j} instead of $\{\bar{j}\}$. For functions $f : \mathbb{R}^d \rightarrow \mathbb{R}$, we often write $E(f)$ instead of $E(f(X))$ for better readability, likewise for Var and Cov .

By a **Generalized Additive Model (GAM)** we mean a function $g : \mathbb{R}^d \rightarrow \mathbb{R}$, $g(X) = g_1(X_1) + \dots + g_d(X_d)$ that can be written as a sum of functions depending on only one feature (Hastie and Tibshirani, 1986). We use the term **Generalized Groupwise Additive Model (GGAM)** in X_S and X_T , where $S, T \subseteq D, S \cap T = \emptyset$, for a function that can be written in the form $g : \mathbb{R}^d \rightarrow \mathbb{R}$, $g(X) = g_S(X_S) + g_T(X_T)$. We refer to the component functions of a GAM (GGAM) as **main effects**. A function that cannot be written as a GAM (GGAM in X_S and X_T) is called an **interaction** (interaction between X_S and X_T).

3.2 Loss-Based Feature Importance

By loss-based feature importance we mean methods that quantify the relevance of features by comparing

the predictive power of subsets of features with respect to some loss (Breiman, 2001; Strobl et al., 2008; Lei et al., 2018; Covert et al., 2020; König et al., 2021; Williamson et al., 2021). To measure the predictive power of a set of features for a specific model f and loss L , we follow the notation of Covert et al. (2020) and introduce a **value function** $v_{f,L} : \mathcal{P}(D) \rightarrow \mathbb{R}_{\geq 0}$, which measures the drop in risk when knowing X_S compared to having access to none of the features. More formally,

$$v_{f,L}(S) := E(L(f_\emptyset, Y)) - E(L(f_S(X_S), Y)),$$

where f_S is a **restricted function** that only has access to the features $S \subseteq D$. Two definitions for the restricted function f_S are common in the literature: the marginal and the conditional version. Both versions integrate out the unused features \bar{S} , but while the **conditional** version integrates over the conditional distribution $P(X_{\bar{S}} | X_S = x_S)$, that is

$$f_S(x_S) := E(f(x_S, X_{\bar{S}}) | X_S = x_S),$$

the **marginal** version integrates over the marginal distribution $P(X_{\bar{S}})$ of the features $X_{\bar{S}}$.

The marginal version is unsuitable for learning about the DGP, since it breaks any dependencies between X_S and $X_{\bar{S}}$ and thereby evaluates the model on unrealistic data (Freiesleben et al., 2024). Thus, we always rely on the conditional version.

We note that in the context of additive models like $g = g_S + g_{\bar{S}}$, the term g_S always refers to the model component and *never* to the restricted function $E(g(X) | X_S = x_S)$; in general they do not coincide. Focusing on regression and on understanding a DGP rather than a particular predictor, we study the \mathcal{L}^2 -loss of the optimal predictor $f^*(x) = E(Y | X = x)$. In this setting, we drop the indices of the value function. It can be shown that the respective value function satisfies

$$\begin{aligned} v(S) &:= v_{f^*, \mathcal{L}^2}(S) = \text{Var}(Y) - E((Y - f_S^*(X_S))^2) \\ &= \text{Var}(E(Y | X_S)), \end{aligned}$$

corresponding to the explained variance of Y conditional on X_S (Covert et al., 2020, Appendix C). We call $v(S)$ the **predictive power** of the features S . We denote the normalized version as $\bar{v}(S) := v(S) / \text{Var}(Y)$. In this setting, f_S^* , based on conditional expectation, can also be estimated by refitting the model with just X_S (Lei et al., 2018; Williamson et al., 2021). The **Leave-One-Covariate-Out (LOCO)** method (Lei et al., 2018; Williamson et al., 2021) uses refitting to compute $v(D) - v(\bar{j})$, which is the drop in predictive power when removing feature j from D .

4 PREDICTIVE POWER DOES NOT REVEAL COOPERATION

Throughout the paper, we develop a method that explains the relevance of cooperation via interactions and dependencies for the predictive power. In this section, we show that even access to the predictive power of all subsets of features – the basis of feature importance methods – is not sufficient to solve this task.

We start the section by introducing what we call the *cooperative impact*, that is, the effect of cooperations on the predictive power. We show that the cooperative impact results from two forces, interactions and dependencies, but that the cooperative impact may not reveal their relevance since their effects on the predictive power might cancel out. Later in the paper we show how to estimate pure interactions in an ML model (Section 5), which will allow us to disentangle the two cooperative forces in the cooperative impact (Section 6).

The Impact of Cooperations On Predictive Power. We start by defining the cooperative impact.

Definition 1 (Cooperative Impact). Let $(X, Y) \sim P$ be a DGP on $\mathbb{R}^d \times \mathbb{R}$ and $J \subseteq D$ a subset of features. The cooperative impact Ψ of J and \bar{J} is defined as

$$\Psi(J, \bar{J}) := v(J \cup \bar{J}) - (v(J) + v(\bar{J})).$$

The cooperative impact results from *two cooperative forces*: interactions and dependencies between the features. As the housing price example in the introduction showed, interactions and dependencies can affect the joint predictive power. They can unlock joint predictive information that is otherwise unavailable (positive cooperative impact) or induce redundancies that reduce the joint contribution (negative cooperative impact). However, in their absence, the joint contribution is simply the sum of the features’ standalone contributions, and the cooperative impact is zero (Proposition 2, proof in Appendix A.1.4).

Proposition 2 (Without Interactions and Dependencies, the Cooperative Effect is Zero). Let $(X, Y) \sim P$ be a DGP and $J \subseteq D$ a subset of features. If X_J and $X_{\bar{J}}$ are independent and the $\mathcal{L}^2(P)$ -optimal predictor can be written as a GGAM $g^* = g_J^* + g_{\bar{J}}^*$ in X_J and $X_{\bar{J}}$, then $\Psi(J, \bar{J}) = 0$.

The Cooperative Impact May Not Reveal the Importance of Cooperations. Although the cooperative impact is zero if no interactions or dependencies are present, the converse is not true. We illustrate this issue with an example.

Example 3 (Contributions of Interactions and Dependencies Cancel Out). Let $Y = X_1 +$

$X_2 + cX_1X_2$, where $X = (X_1, X_2)$ is normally distributed on \mathbb{R}^2 with $\text{Var}(X_1) = \text{Var}(X_2) = 1$ and $\text{Cov}(X_1, X_2) = \beta$.

In DGP 1, we set $c = \beta = 0$, meaning there are no interactions or dependencies, and thus no cooperations. In DGP 2, we set $c = \sqrt{6}$ and $\beta = 0.5$ such that there is cooperation both in the form of interactions and dependencies. In both cases we obtain $\bar{v}(1 \cup 2) = 1$, $\bar{v}(1) = 0.5$, and $\bar{v}(2) = 0.5$. Hence $\Psi(1, 2) = 0$. See Appendix A.2.1 for the formal derivation.

In the example, there is no cooperation in the first DGP, but the variables cooperate both in form of interactions and dependencies in the second DGP. Nevertheless, the value functions for all possible sets of features are the same in both DGPs; the cooperative impact is zero. The reason is that in the second DGP the positive effect of the interaction and the negative effect of the redundancy-inducing dependence cancel each other out.

The example shows that value functions are not sufficient to quantify the relevance of interactions and dependencies. To reveal their impact, we disentangle them by first decomposing the prediction model.

5 ESTIMATING PURE INTERACTIONS

In this section, our goal is to separate interactions and main effects in an ML model f . Given two groups of variables, we decompose f into an *interaction term* that only represents interactions *between* the groups, and *main effects* that only permit interactions *within* groups.

Characterizing Pure Interactions Using Additive Models More formally, given two groups of features J and \bar{J} , we want to decompose a prediction function f as

$$f(x) = g_J(x_J) + g_{\bar{J}}(x_{\bar{J}}) + h(x),$$

where g is the model that only permits within-group interactions, that is, a generalized groupwise additive model (GGAM) of the form $g = g_J + g_{\bar{J}}$. Moreover, we want the remaining interaction term h to be “pure”, meaning that everything that can be expressed without between-group interactions should be represented in the GGAM g . This intuition gives rise to the following definition.

Definition 4 (Pure Interaction). Let P be a probability distribution on \mathbb{R}^d , $f : \mathbb{R}^d \rightarrow \mathbb{R}$ a function and $J \subseteq D$ a subset of features. Let further g be the $\mathcal{L}^2(P)$ -optimal approximation of f within all GGAMs in X_J and $X_{\bar{J}}$. We define the *pure interaction* of f with respect to X_J and $X_{\bar{J}}$ as $f - g$.

In short, the pure interaction represents what cannot be explained by a GGAM. Since the $\mathcal{L}^2(P)$ -optimal predictor of a topologically closed function class is unique, the pure interaction h is unique as well.

This characterization of pure interactions between groups is not only unique and intuitive; in the following paragraphs we show that it entails a simple estimation procedure and overlaps with existing definitions. As we will see in Section 6, its properties allow finding an interpretable decomposition of the predictive power.

Estimation Our definition of pure interactions directly entails a procedure for their estimation. To find the pure interaction in f with respect to two groups J and \bar{J} , we only need to fit one GGAM g approximating f to get the pure interaction as the residual $h = f - g$. Thereby, we can leverage a broad range of machine learning methods and implementations (Hastie and Tibshirani, 1986; Servén and Brummitt, 2018; Nori et al., 2019).

In our context, we want to decompose the loss-optimal model f^* , and therefore approximate f^* with a GGAM. As Lemma 11 in Appendix A.1.1 shows, we can equivalently approximate Y using the GGAM. Since only an approximation of f^* is available in practice, we fit the GGAM directly on Y in our experiments.

Properties and Relation To Existing Definitions

We can equivalently characterize pure interactions as terms that are not approximable by only one of the two groups of variables, as the following theorem shows (proof in Appendix A.1.2).

Theorem 5 (Equivalent Characterization of Pure Interactions). *Let P be a probability distribution on \mathbb{R}^d , $f : \mathbb{R}^d \rightarrow \mathbb{R}$ a function and $J \subseteq D$ a subset of features. A GGAM $g = g_J + g_{\bar{J}}$ is the $\mathcal{L}^2(P)$ -optimal approximation of f within all GGAMs in X_J and $X_{\bar{J}}$ if and only if the residual $h := f - g$ satisfies $E(h | X_J) = 0$ and $E(h | X_{\bar{J}}) = 0$.*

Theorem 5 enables another interpretation of pure interactions. In particular, $E(h | X_J) = E(h | X_{\bar{J}}) = 0$ implies that pure interactions in the optimal predictor f^* for a DGP do not contribute to the standalone predictive powers $v(J)$ and $v(\bar{J})$.

Furthermore, Theorem 5 implies that in the two-dimensional setting, our definition of pure interactions coincides with the one in Lengerich et al. (2020), which is based on Hooker (2007).

We highlight that pure interactions depend not only on the function f but also on the dependencies between features; see Appendix A.2.1 for an example.

Uniqueness of the GGAM Components In addition to the uniqueness of the pure interaction

term h , we show that under mild assumptions the component functions g_J and $g_{\bar{J}}$ of the GGAM g in Definition 4 are unique up to a constant, see Appendix A.1.3. This ensures that our decomposition $f = g_J + g_{\bar{J}} + h$ is unique up to a constant as well, which will later imply the uniqueness of the DIP decomposition.

The uniqueness of the GGAM components breaks, for example, when the variables X_J and $X_{\bar{J}}$ are perfectly correlated such that any function in X_J can also be written in terms of $X_{\bar{J}}$. Thus, intuitively speaking, our assumption assures that all combinations of features can occur in the distribution. More formally, if the features are discrete we require all combinations to have positive probability; if they are continuous we require the density to be strictly positive everywhere and the component functions to be continuous or to take finitely many values. We note that these assumptions are met by most common distributions, such as non-degenerate multivariate normal distributions, and most ML models, such as tree-based models or neural networks.

6 DISENTANGLING THE CONTRIBUTIONS OF INTERACTIONS AND DEPENDENCIES

Equipped with the tools to decompose a model f , we are ready to disentangle the effects of interactions and dependencies on the predictive power (DIP decomposition). More precisely, we decompose the cooperative impact (see Definition 1) in Theorem 6. For the proof we refer to Appendix A.1.4.

Theorem 6 (Cooperative Impact Decomposition). *Let $(X, Y) \sim P$ be a DGP on $\mathbb{R}^d \times \mathbb{R}$, $J \subseteq D$ a subset of features, $f^* = E(Y | X)$ the $\mathcal{L}^2(P)$ -optimal predictor and $g^* = g_J^* + g_{\bar{J}}^*$ the $\mathcal{L}^2(P)$ -optimal GGAM in X_J and $X_{\bar{J}}$. We call $h^* := f^* - g^*$. Then, we get a decomposition*

$$\begin{aligned} \Psi(J, \bar{J}) &= \underbrace{\text{Var}(h^*)}_{\text{Interaction Surplus}} - \underbrace{\text{Dep}(J, \bar{J})}_{\text{Main Effect Dependencies}}, \quad \text{where} \\ \text{Dep}(J, \bar{J}) &:= \underbrace{\text{Var}(E(g_J^* | X_J)) + \text{Var}(E(g_{\bar{J}}^* | X_{\bar{J}}))}_{\text{Cross-Predictability}} \\ &\quad + \underbrace{2 \text{Cov}(g_J^*, g_{\bar{J}}^*)}_{\text{Covariance}}. \end{aligned}$$

Interpretation First, we recall that both pure interactions and main effects are unique up to a constant under mild assumptions (Section 5, Appendix A.1.3), and thus the decomposition is unique, too.

Next, we highlight that the cooperative impact decomposes into the sum of a term that only depends on the pure interaction h^* and terms that only depend on the main effects g^* . In other words, the contributions of interactions and main effects simply add up. This is no coincidence but a direct consequence of the properties of pure interactions (Definition 4). These properties ensure that any covariance terms involving h and g vanish (cf. the proof of Theorem 6).

The effect of interactions on the predictive power is given by the variance of the pure interaction term h^* . Since the variance cannot be negative, we refer to it as the **interaction surplus**. It measures how much of the joint predictive power can only be explained with between-group interactions.

We notice that $\text{Dep}(J, \bar{J})$ vanishes if the two groups of features X_J and $X_{\bar{J}}$ are independent. It measures how much of the cooperative impact is caused by *dependencies*. More precisely, by the dependencies between the effects of X_J and $X_{\bar{J}}$ on Y , given by the main effects g_J^* and $g_{\bar{J}}^*$. Thus we refer to $\text{Dep}(J, \bar{J})$ as the **main effect dependencies**. As we will see, the main effect dependencies can have a positive or negative influence on the cooperative impact. The main effect dependencies consist of two parts: We refer to them as the main effect cross-predictability and the main effect covariance.

Intuitively, the **main effect cross-predictability** quantifies how redundant the contributions of the two groups of features are. More formally, it measures how much of the variance of each main effect could also be explained by the respective other group of variables. Being a sum of variances, the cross-predictability is always positive, and its impact on the cooperative impact is always negative. This is consistent with the intuition that the joint predictive power should decrease if the variables share more variation.

Unlike the cross-predictability, the **main effect covariance** can be either positive or negative. If it is negative and its absolute value outweighs the cross-predictability, the main effect dependencies increase the cooperative impact $\Psi(J, \bar{J})$. This may seem counterintuitive because then the two groups of variables are more predictive together than individually, even if they do not interact. For an intuitive example on how a negative main effect covariance induces this improvement in predictive power, we refer to Example 8 below. This phenomenon can also be observed in multivariate linear models and is tied to the terms “enhancement” (Friedman and Wall, 2005) or “suppression” (Shieh, 2006). We extend the analysis of this phenomenon to the more general setting of ML involving GAMs.

Note that $\text{Dep}(J, \bar{J})$ and its two components cannot simply be determined from the dependencies between features. Instead they explain the *relevance* of dependencies

for the underlying target, measured by the predictive power (details in Appendix A.2.2 and A.2.3).

Estimation Irrespective of the dimensionality of J and \bar{J} we need access to three models to compute the cooperative impact $\Psi(J, \bar{J})$: the full model f^* and the restricted models f_J^* and $f_{\bar{J}}^*$. To decompose the cooperative impact using DIP, only one additional model fit is required (the GGAM g^*). Since the number of model fits needed to compute one DIP decomposition is independent of the dimension of the underlying space, DIP can also be applied in higher dimensional settings.

In practice, the optimal models are not available but can be approximated using ML. To avoid bias due to overfitting when estimating the components of the DIP decomposition, the scores can be reformulated in terms of empirical risk on test data (details in Appendix B.1).

Illustrative Examples First, we revisit Example 3, where we can now reveal the relevance of cooperation (derivations in Appendix A.2.1). Then we illustrate the interpretation of DIP in Examples 7 to 9 (derivations in Appendix A.2.4).

Example 3 Continued (Cooperative Forces May Cancel Out, Figure 2a). Using DIP, we can distinguish the cooperative and non-cooperative DGP in Example 3. For DGP 1 ($c = \beta = 0$), the interaction surplus and the main effect dependencies are both zero, as one would expect; see Figure 2a, left bar. On the other hand, for DGP 2 ($c = \sqrt{6}, \beta = 0.5$) we get a (normalized) interaction surplus of 0.26, cross-predictability of 0.07 and covariance of 0.19, such that interaction surplus and main effect dependencies cancel out (Figure 2a, right bar).

Example 7 (Negative Cooperative Impact via Dependence, Figure 2b). Suppose we want to predict a student’s points Y in an exam based on two binary features that indicate whether a student did their homework ($X_1 = 1$) or not ($X_1 = 0$) and whether a student studied for at least 10 hours ($X_2 = 1$) or not ($X_2 = 0$). We assume the two features are $\text{Ber}(0.5)$ -distributed and positively correlated with $P(X_1 = X_2) = 0.75$ (see Figure 2b, top left), because doing homework requires time. Assume the points follow the rule $Y = 4X_1 + 4X_2$ (see Figure 2b, bottom left), meaning there are no interactions, so $h^* = 0$. This results in a negative cooperative impact $\Psi(1, 2) = -6$ ($v(1) = 9$, $v(2) = 9$, and $v(1 \cup 2) = 12$). The DIP decomposition delivers a **cross-predictability of 2** and a **main effect covariance of 4**, whose negatives sum up to the observed cooperative effect of $\Psi(1, 2) = -6$.

Intuitively, this negative cooperative impact makes sense since the two features are correlated and can

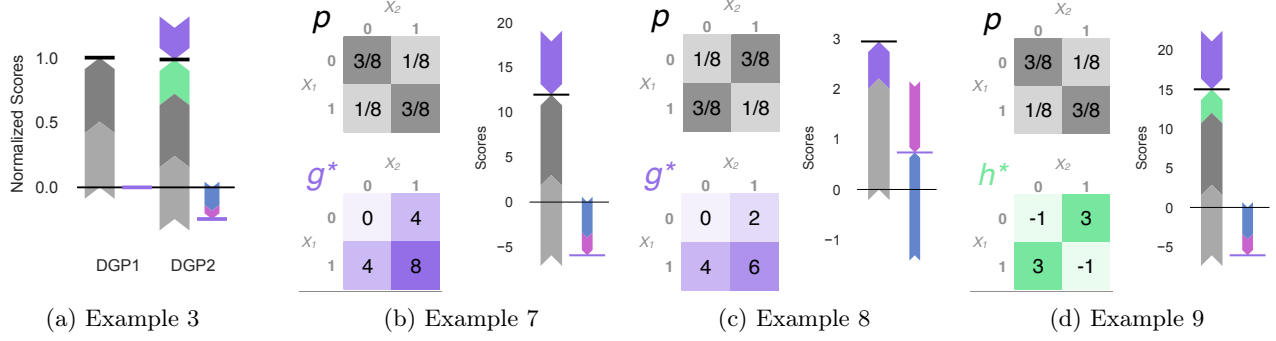


Figure 2: **Examples.** For each example, we show a forceplot visualizing the DIP decomposition into standalone contributions ($v(1)$ and $v(2)$), **main effect dependencies** (Dep(1, 2)) and **interaction surplus**, where the direction of each bar (upward or downward) represents the sign. They sum up to $v(1, 2)$ (black horizontal line). The slim bars (right) show the decomposition of Dep(1, 2) (purple horizontal line) into **covariance** and **cross-predictability**. For Examples 7-9, we additionally show heatmaps visualizing the distribution (top) and g^* or h^* (bottom).

partially replace each other. This leads to either variable being able to recover more than half of the explained variance of Y on its own and thus $\Psi(1, 2) < 0$. In short, the negative effect of the main effect dependencies indicates that the two variables have similar information about the target.

Example 8 (Positive Cooperative Impact via Dependence, Figure 2c). In our second example, Y still reflects the points and X_1 whether a student did homework, but feature 2 now indicates whether the student attended the review session ($X_2 = 1$) or not ($X_2 = 0$). Students who did their homework tend not to attend the review session, so we again choose X_1 and X_2 to be Ber(0.5)-distributed, but this time negatively correlated with $P(X_1 = X_2) = 0.25$, see Figure 2c, top left. The relationship between points and features is given by $Y = 4X_1 + 2X_2$ (see Figure 2c, bottom left), so again $h^* = 0$. This time, we get a positive cooperative impact ($v(1) = 2.25$, $v(2) = 0$, and $v(1 \cup 2) = 3$). The **cross-predictability is 1.25** and the **main effect covariance** -2 , outweighing the former.

Why do the two features have more predictive power together than individually, despite the absence of any interaction? Although attending the review session adds two points, we have $v(2) = 0$. Indeed, both columns of the lower table in Figure 2c have a weighted average of 3, that is, $E(Y | X_2 = 0) = E(Y | X_2 = 1) = 3$. So, knowing solely X_2 does not help predicting Y . This is due to the correlation. For students who attended the review session, it is more likely that they did not do homework, which is bad for their score. This cancels out the positive effect of the review session. One could say the correlation works against the prediction. Once we use both features, the predictive power of X_2 is revealed because X_1 is known. The lower table of Figure 2c again illustrates this well:

Once we can distinguish the rows, the difference between the two columns becomes visible.

The crucial part of Example 8 is that the values of X_1 and X_2 that are likely to occur concurrently have opposing effects on Y . Formally, that is $\text{Cov}(g_1^*, g_2^*) < 0$. The example illustrates how a negative main effect covariance can improve the features' joint predictive power compared to their individual ones.¹

Example 9 (Interactions, Figure 2d). Let us again consider the positively correlated Ber(0.5)-distributed variables X_1 and X_2 from Example 7 that satisfy $P(X_1 = X_2) = 0.75$. This time, we set $Y = 8(X_1 \vee X_2) - 1$, where \vee denotes the logical OR-operator. This may be rewritten as $Y = 4X_1 + 4X_2 + 4(X_1 \oplus X_2) - 1$, where \oplus denotes the XOR-operator. The expression $h(X) = 4(X_1 \oplus X_2) - 1$ is a pure interaction. To verify this, consider the lower table of Figure 2d. The weighted average of each row and each column is zero, which formally means $E(h | X_1) = 0$ and $E(h | X_2) = 0$.² Consequently, we obtain the (unique) decomposition of Y into the GAM $g^*(X) = 4X_1 + 4X_2$ and the pure interaction $h^*(X) = 4(X_1 \oplus X_2) - 1$. Using this decomposition, we get the same standalone and main effect dependence contributions as in Example 7 ($v(1) = 9$, $v(2) = 9$, **cross-predictability of 2**, and **covariance of 4**). This illustrates that the standalone contributions and Dep(J, \bar{J}) are not affected by the pure interaction. The **interaction surplus** $\text{Var}(h^*) = 3$ is simply added to these values causing $\Psi(1, 2)$ and $v(1 \cup 2)$ to be increased by 3.

¹Note that this phenomenon does not need one of the two value functions to vanish. We simply chose an example with a vanishing value function for better illustration.

²Note that the constant -1 is required to ensure that h is mean-centered, which is a necessary condition for our definition. Otherwise, the expressions $E(h | X_1)$ and $E(h | X_2)$ would be constant but not zero.

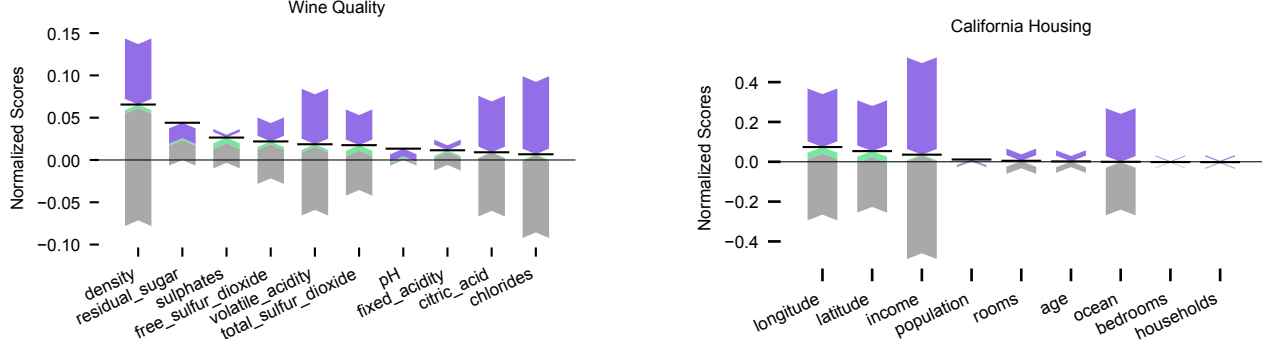


Figure 3: **Applications.** We decompose the LOCO scores on the wine quality dataset (left) and the California Housing dataset (right) into each feature’s standalone contribution, the **interaction surplus**, and the contribution of **main effect dependencies**.

7 APPLYING THE DECOMPOSITION TO FEATURE IMPORTANCE

Now, we show how to apply the DIP decomposition to explain Leave-One-Covariate-Out (LOCO) importance (Lei et al., 2018; Williamson et al., 2021), a popular feature importance technique. The LOCO score is defined as the drop in predictive power when removing one variable from the full set of features and can be rewritten using Definition 1 as

$$LOCO_j := v(j \cup \bar{j}) - v(\bar{j}) = v(j) + \Psi(j, \bar{j}).$$

To explain the relevance of interactions and dependencies for a LOCO score, we can decompose the cooperative impact $\Psi(j, \bar{j})$ using Theorem 6.

Using the same trick, we can also explain the relevance of cooperation for Shapley effects (Song et al., 2016), also called SAGE values (Covert et al., 2020) (Appendix A.3). More generally, we can apply the method to any feature importance method based on predictive power comparisons of the form $v(S \cup T) - v(T)$.

Notably, computing the DIP decomposition of a feature importance method does not increase the asymptotic complexity of the method, since a constant number of additional model fits is needed for the decomposition of each surplus $v(j \cup \bar{j}) - v(\bar{j})$. For details on runtime we refer the reader to Appendix B.2.

8 APPLICATIONS

We now apply the DIP decomposition to real-world data. More specifically, we compute and decompose LOCO scores. We apply the method to two datasets, the wine quality dataset (Cortez et al., 2009) and the California Housing dataset found in (Géron, 2022).

Implementation To estimate the full model and to decompose it into pure interactions and main ef-

fects, we leverage the explainable boosting machine implemented in the `interpretML` package (Nori et al., 2019). We compute the scores on test data as described in Appendix B.1. We employ a 10-fold cross-validation scheme. All scores are normalized by the variance of the target variable, thus indicating the proportion of the variance that is explained. We implemented the methods as a `python` package; all code is publicly available on GitHub. Details on the implementation and computational cost are reported in Appendix B.2.

Wine Quality For this dataset ($n = 6496$, Cortez et al. (2009)), obtained from the UCI ML Repository (Dua and Graff, 2017) the goal is to predict *wine quality* (a score between one and ten) using ten physico-chemical characteristics such as *citric acidity*, *residual sugar*, and *density*. Suppose we use LOCO to gain insight into which variables are most relevant for predicting the target. The scores in Figure 3 (left, black horizontal lines) suggest that *density* and *residual sugar* are most relevant, but that *citric acidity* is irrelevant. The scores are prone to misinterpretation.

First, one may erroneously infer that variables with large scores, like *residual sugar*, also have large standalone predictive power. The DIP decomposition (Figure 3, left) reveals that *residual sugar* is relevant due to cooperation instead, and thus, its role in predicting the target can only be understood in combination with other features. A pairwise DIP decomposition further reveals a positive cooperative impact between *residual sugar* and *density* (Appendix C.1.1); an in-depth analysis shows that the features are positively correlated (adding sugar increases the density) but have opposing effects on wine quality that cancel out unless both features are observed (Appendix C.1.2).

Second, one may erroneously conclude that features with small LOCO scores, like *citric acidity*, contain

little predictive information about Y . Instead, DIP reveals that *citric acidity* is one of the most predictive standalone features but is considered irrelevant by LOCO due to its redundancy with the remaining features. A pairwise DIP decomposition reveals that *citric acidity* shares most of its contribution with *volatile acidity*, suggesting that they have similar roles for the target (Appendix C.1.1).

California Housing The goal of the California Housing dataset ($n = 20433$, Géron (2022)) is to predict the 1990 median house price of districts in California based on characteristics such as *longitude*, *latitude*, and *ocean proximity*. The variables *longitude* and *latitude* have the highest LOCO scores, and one may erroneously conclude that the variables are individually important for the outcome. However, as DIP reveals, the variables’ LOCO scores mostly stem from interactions (Figure 3, right). As such, we need to consider both variables together to fully understand their role for Y . Furthermore, DIP reveals that seemingly irrelevant variables such as *ocean proximity* and *income* in fact are useful standalone predictors that receive low scores since they share their contributions with the remaining features.

Additional experiments are reported in Appendix C. We use pairwise DIP decompositions to better understand which features cooperate, we apply the DIP decomposition to another feature importance technique called SAGE, and we demonstrate the usefulness of DIP on two higher-dimensional datasets. In Appendix D we demonstrate that the DIP decomposition can be used to assess whether the cPDP (Apley and Zhu, 2020), a technique explaining the effect of each feature on the prediction, can be trusted to faithfully reflect the role of a feature in a multivariate context.

9 CONCLUSION

Throughout the paper, we introduced DIP, a method to decompose the \mathcal{L}^2 -loss-based predictive power of two groups of features into their standalone predictive power as well as their cooperation via interactions and dependencies.

DIP can be used to explain the outputs of commonly used global feature importance methods such as LOCO (Lei et al., 2018; Williamson et al., 2021) or SAGE (Covert et al., 2020). More generally, DIP is applicable to any method based on predictive power comparisons involving two groups of features.

Thereby, DIP allows novel insight into the DGP: It reveals which features are individually relevant and which are due to interactions and dependencies, showing whether variables must be analyzed jointly to understand their role for the target. Furthermore, we

can assess which variables share predictive contributions and thus have similar relationships with Y ; insight that cannot be obtained by simply measuring the dependencies between features. Since these questions are of central interest to scientific inquiry, we are convinced that DIP has great potential to enable relevant discoveries.

Acknowledgements

The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Eric Günther. This work has been supported by the German Research Foundation through the Cluster of Excellence “Machine Learning - New Perspectives for Science” (EXC 2064/1 number 390727645). The authors thank Sebastian Bordt, Moritz Haas, and Karolin Frohnapfel for their valuable feedback, Giles Hooker for an email exchange on the uniqueness of the fANOVA decomposition, and Moritz Grosse-Wentrup for the insightful discussions that helped inspire this project.

References

- Apley, D. W. and Zhu, J. (2020). Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 82(4):1059–1086.
- Barrett, A. B. (2015). Exploration of synergistic and redundant information sharing in static and dynamical gaussian systems. *Physical Review E*, 91(5):052802.
- Bordt, S. and von Luxburg, U. (2023). From Shapley values to generalized additive models and back. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.
- Breiman, L. (2001). Random forests. *Machine learning*, 45:5–32.
- Chastaing, G., Gamboa, F., and Prieur, C. (2015). Generalized sobol sensitivity indices for dependent variables: numerical methods. *Journal of statistical computation and simulation*, 85(7):1306–1333.
- 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*.
- Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision support systems*, 47(4):547–553.
- Covert, I., Lundberg, S., and Lee, S.-I. (2021). Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research*, 22(209):1–90.

- Covert, I., Lundberg, S. M., and Lee, S.-I. (2020). Understanding global feature contributions with additive importance measures. *Neural Information Processing Systems (NeurIPS)*, 33.
- Dua, D. and Graff, C. (2017). UCI machine learning repository.
- Fernandes, K., Vinagre, P., and Cortez, P. (2015). A proactive intelligent decision support system for predicting the popularity of online news. In *Progress in Artificial Intelligence: 17th Portuguese Conference on Artificial Intelligence, EPIA 2015, Coimbra, Portugal, September 8-11, 2015. Proceedings 17*, pages 535–546. Springer.
- Freiesleben, T., König, G., Molnar, C., and Tejero-Cantero, Á. (2024). Scientific inference with interpretable machine learning: Analyzing models to learn about real-world phenomena. *Minds and Machines*, 34(3):32.
- Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.
- Friedman, J. H. and Popescu, B. E. (2008). Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3):916–954.
- Friedman, L. and Wall, M. (2005). Graphical views of suppression and multicollinearity in multiple linear regression. *The American Statistician*, 59(2):127–136.
- Fumagalli, F., Muschalik, M., Kolpaczki, P., Hüllermeier, E., and Hammer, B. (2024a). KernelSHAP-IQ: Weighted least-square optimization for Shapley interactions. *arXiv preprint arXiv:2405.10852*.
- Fumagalli, F., Muschalik, M., Kolpaczki, P., Hüllermeier, E., and Hammer, B. (2024b). SHAP-IQ: Unified approximation of any-order Shapley interactions. *Neural Information Processing Systems (NeurIPS)*.
- Gao, Y., Sahin, A., and Vrugt, J. A. (2023). Probabilistic sensitivity analysis with dependent variables: Covariance-based decomposition of hydrologic models. *Water Resources Research*, 59(4):e2022WR032834.
- Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. O’Reilly Media, Inc.
- Grabisch, M. and Roubens, M. (1999). An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of game theory*, 28:547–565.
- Griffith, V. and Ho, T. (2015). Quantifying redundant information in predicting a target random variable. *Entropy*, 17(7):4644–4653.
- Hamidieh, K. (2018). A data-driven statistical model for predicting the critical temperature of a superconductor. *Computational Materials Science*, 154:346–354.
- Hastie, T. and Tibshirani, R. (1986). Generalized additive models. *Statist. Sci.*, 1(4):297–310.
- Herbinger, J., Bischl, B., and Casalicchio, G. (2023). Decomposing global feature effects based on feature interactions. *arXiv preprint arXiv:2306.00541*.
- Herren, A. and Hahn, P. R. (2022). Statistical aspects of shap: Functional anova for model interpretation. *arXiv preprint arXiv:2208.09970*.
- Hiabu, M., Meyer, J. T., and Wright, M. N. (2023). Unifying local and global model explanations by functional decomposition of low dimensional structures. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.
- Hooker, G. (2007). Generalized functional anova diagnostics for high-dimensional functions of dependent variables. *Journal of computational and graphical statistics*, 16(3):709–732.
- Hooker, G., Mentch, L., and Zhou, S. (2021). Unrestricted permutation forces extrapolation: variable importance requires at least one more model, or there is no free variable importance. *Statistics and Computing*, 31:1–16.
- Janizek, J. D., Sturmfels, P., and Lee, S.-I. (2021). Explaining explanations: Axiomatic feature interactions for deep networks. *JMLR*, 22(104):1–54.
- Kolchinsky, A. (2022). A novel approach to the partial information decomposition. *Entropy*, 24(3):403.
- König, G., Molnar, C., Bischl, B., and Grosse-Wentrup, M. (2021). Relative feature importance. In *International Conference on Pattern Recognition (ICPR)*.
- Lei, J., G’Sell, M., Rinaldo, A., Tibshirani, R. J., and Wasserman, L. (2018). Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111.
- 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 (AISTATS)*.
- Li, G. and Rabitz, H. (2012). General formulation of hdmr component functions with independent and correlated variables. *Journal of mathematical chemistry*, 50:99–130.
- Luenberger, D. G. (1997). *Optimization by vector space methods*. John Wiley & Sons.

- Lundberg, S. M., Erion, G. G., and Lee, S.-I. (2018). Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*.
- Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Neural Information Processing Systems (NeurIPS)*.
- Masoomi, A., Hill, D., Xu, Z., Hersh, C. P., Silverman, E. K., Castaldi, P. J., Ioannidis, S., and Dy, J. (2022). Explanations of black-box models based on directional feature interactions. In *International Conference on Learning Representations (ICLR)*.
- Molnar, C. (2022). *Interpretable Machine Learning*. 2. edition.
- Muschalik, M., Fumagalli, F., Hammer, B., and Hüllermeier, E. (2024). Beyond treeSHAP: Efficient computation of any-order Shapley interactions for tree ensembles. In *AAAI Conference on Artificial Intelligence*.
- Nathans, L. L., Oswald, F. L., and Nimon, K. (2012). Interpreting multiple linear regression: a guidebook of variable importance. *Practical assessment, research & evaluation*, 17(9):n9.
- Nori, H., Jenkins, S., Koch, P., and Caruana, R. (2019). Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*.
- Ray-Mukherjee, J., Nimon, K., Mukherjee, S., Morris, D. W., Slotow, R., and Hamer, M. (2014). Using commonality analysis in multiple regressions: a tool to decompose regression effects in the face of multicollinearity. *Methods in Ecology and Evolution*, 5(4):320–328.
- Seibold, D. R. and McPhee, R. D. (1979). Commonality analysis: A method for decomposing explained variance in multiple regression analyses. *Human Communication Research*, 5(4):355–365.
- Servén, D. and Brummitt, C. (2018). pygam: Generalized additive models in python. *Zenodo*.
- Shieh, G. (2006). Suppression situations in multiple linear regression. *Educational and psychological measurement*, 66(3):435–447.
- Song, E., Nelson, B. L., and Staum, J. (2016). Shapley effects for global sensitivity analysis: Theory and computation. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):1060–1083.
- Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., and Zeileis, A. (2008). Conditional variable importance for random forests. *BMC bioinformatics*, 9(307):1–11.
- Sundararajan, M., Dhamdhere, K., and Agarwal, A. (2020). The Shapley Taylor interaction index. In *International conference on machine learning (ICML)*.
- Williams, P. L. and Beer, R. D. (2010). Nonnegative decomposition of multivariate information. *arXiv preprint arXiv:1004.2515*.
- Williamson, B. D., Gilbert, P. B., Carone, M., and Simon, N. (2021). Nonparametric variable importance assessment using machine learning techniques. *Biometrics*, 77(1):9–22.
- Withers, C. S. (1985). The moments of the multivariate normal. *Bulletin of the Australian Mathematical Society*, 32(1):103–107.
- Zhang, H., Xie, Y., Zheng, L., Zhang, D., and Zhang, Q. (2021). Interpreting multivariate shapley interactions in dnns. In *AAAI*.

Checklist

1. For all models and algorithms presented, check if you include:
 - (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes, see Section 3.]
 - (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not applicable.]
 - (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes, for details refer to Appendix B.2.]
2. For any theoretical claim, check if you include:
 - (a) Statements of the full set of assumptions of all theoretical results. [Yes.]
 - (b) Complete proofs of all theoretical results. [Yes, see Appendix A.]
 - (c) Clear explanations of any assumptions. [Yes]
3. For all figures and tables that present empirical results, check if you include:
 - (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes, see Appendix B.]
 - (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes, see Appendix B.]
 - (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes, see Appendix B.]
 - (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes, see Appendix B.]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
 - (a) Citations of the creator If your work uses existing assets. [Yes.]
 - (b) The license information of the assets, if applicable. [Not Applicable]
 - (c) New assets either in the supplemental material or as a URL, if applicable. [Not applicable.]
 - (d) Information about consent from data providers/curators. [Not Applicable.]
 - (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable.]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
 - (a) The full text of instructions given to participants and screenshots. [Not Applicable.]
 - (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable.]
 - (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable.]

A THEORY

A.1 Proofs

A.1.1 Necessary Lemmata

Lemma 10 (Equivalence of Orthogonality and Non-approximability). *Let P be a probability distribution on \mathbb{R}^d , $f : \mathbb{R}^d \rightarrow \mathbb{R}$ a function and $J \subseteq \mathbb{R}$ a subset of variables. Then the following are equivalent:*

1. $E(f \cdot h) = 0$ for all $h \in \mathcal{L}^2(\mathbb{R}^J, P)$
2. $E(f \mid X_J) = 0$.

Proof.

1. \implies 2. Note that by setting $h \equiv 1$ we get $E(f) = 0$. We start by considering $h = E(f \mid X_J)$ which gives $E(f \cdot E(f \mid X_J)) = 0$. Since f is centered we can replace the expected value of the product by a covariance and then use the law of total covariance to receive

$$\begin{aligned} 0 &= E(f \cdot E(f \mid X_J)) = \text{Cov}(f, E(f \mid X_J)) \\ &= \text{Cov}(E(f \mid X_J), E(f \mid X_J)) + \underbrace{E(\text{Cov}(f, E(f \mid X_J) \mid X_J))}_{=0} \\ &= \text{Var}(E(f \mid X_J)), \end{aligned}$$

so $E(f \mid X_J)$ is constant. Since its expected value yields $E(E(f \mid X_J)) = E(f) = 0$, the function $E(f \mid X_J)$ must already be constant zero.

2. \implies 1. Note that again, $E(f) = E(E(f \mid X_J)) = 0$. For an arbitrary $h \in \mathcal{L}^2(\mathbb{R}^J, P)$, using the law of total covariance, we directly compute

$$E(f \cdot h) = \text{Cov}(f, h) = \underbrace{\text{Cov}(E(f \mid X_J), E(h \mid X_J))}_{=0} + \underbrace{E(\text{Cov}(f, h \mid X_J))}_{=0} = 0.$$

□

Lemma 11 (Equivalence of Approximating the Data and Approximating a Better Predictor). *Let $(X, Y) \sim P$ be a data generating process on $\mathbb{R}^d \times \mathbb{R}$ and consider two function classes \mathcal{F} and \mathcal{G} such that $\mathcal{G} \subseteq \mathcal{F}$. Let f be the $\mathcal{L}^2(P)$ -optimal predictor in the function class \mathcal{F} . Then, for a function $g \in \mathcal{G}$ the following are equivalent:*

1. *The function g is the $\mathcal{L}^2(P)$ -optimal predictor for Y within \mathcal{G}*
2. *The function g is the $\mathcal{L}^2(P)$ -optimal approximation of f within \mathcal{G} .*

Proof. Note that a predictor is $\mathcal{L}^2(P)$ -optimal within a function class if and only if its residual is perpendicular to the function class (Luenberger, 1997).

1. \implies 2. Let us assume that g is optimal for Y . So, for an arbitrary $h \in \mathcal{G}$ we have

$$\begin{aligned} 0 &= E((Y - g(X)) \cdot h(X)) = \underbrace{E((Y - f(X)) \cdot h(X))}_{=0} + E((f(X) - g(X)) \cdot h(X)) \\ &= E((f(X) - g(X)) \cdot h(X)), \end{aligned}$$

where $E((Y - f(X)) \cdot h(X))$ vanishes because f is the optimal predictor for Y within \mathcal{F} and h is contained in \mathcal{F} . From that, we directly conclude that g is $\mathcal{L}^2(P)$ -optimal for $f(X)$.

2. \implies 1. If g is optimal for $f(X)$ we again take an arbitrary $h \in \mathcal{G}$ and compute

$$\begin{aligned} 0 &= E((f(X) - g(X)) \cdot h(X)) = E((Y - g(X)) \cdot h(X)) - \underbrace{E((Y - f(X)) \cdot h(X))}_{=0} \\ &= E((Y - g(X)) \cdot h(X)) \end{aligned}$$

and hence, g is optimal for Y .

□

A.1.2 Proof of Theorem 5

Proof of Theorem 5. The function g is the $\mathcal{L}^2(P)$ -optimal approximation of f within all GGAMs in X_J and $X_{\bar{J}}$ if and only if $f - g$ is perpendicular to the class of all GGAMs in X_J and $X_{\bar{J}}$ (Luenberger, 1997). This is equivalent of saying $f - g$ is perpendicular to all functions that only depend on X_J and all functions that only depend on $X_{\bar{J}}$. The claim then follows from Lemma 10. □

A.1.3 Uniqueness of the GGAM Components

Here, we prove that the GGAM components are unique up to a constant under some mild assumptions on the underlying distribution and the GGAM. Note that a stronger result than uniqueness up to a constant cannot be proven because it is always possible to add a constant to g_J and subtract the same one from $g_{\bar{J}}$, no matter how strong our assumptions are. However, when decomposing the cooperative impact in Theorem 6, we take variances and covariances and hence, these constants do not change the DIP decomposition.

Theorem 12 (Uniqueness of the GGAM Components). *Let P be a probability distribution on \mathbb{R}^d and $g : \mathbb{R}^d \rightarrow \mathbb{R}$ a measurable function that admits a decomposition*

$$g(x) = g_J(x_J) + g_{\bar{J}}(x_{\bar{J}}).$$

Assume that one of the following three assumptions is satisfied:

1. *The probability measure on \mathbb{R}^d is discrete and $P(A_J \times A_{\bar{J}}) > 0$ for all values $A_J, A_{\bar{J}}$ that X_J and $X_{\bar{J}}$ take*
2. *The probability measure on \mathbb{R}^d is continuous with some strictly positive density $p > 0$ and $g, g_J, g_{\bar{J}}$ only take finitely many values*
3. *The probability measure on \mathbb{R}^d is continuous with some strictly positive density $p > 0$ and $g, g_J, g_{\bar{J}}$ are continuous.*

Then, the components g_J and $g_{\bar{J}}$ are unique almost surely up to a constant.

Proof. Given two decompositions $g = g_J + g_{\bar{J}} = \tilde{g}_J + \tilde{g}_{\bar{J}}$ we may subtract them to receive

$$(g_J - \tilde{g}_J) + (g_{\bar{J}} - \tilde{g}_{\bar{J}}) = 0 \quad a.s.$$

For simplicity of the notation we write $f_J := g_J - \tilde{g}_J$ and $f_{\bar{J}} := -(g_{\bar{J}} - \tilde{g}_{\bar{J}})$. So we know $f_J = f_{\bar{J}}$ a.s., from which we will derive that f_J and $f_{\bar{J}}$ are already constant a.s. This is the same as saying g_J and \tilde{g}_J as well as $g_{\bar{J}}$ and $\tilde{g}_{\bar{J}}$ coincide up to a constant.

Lets first consider the assumptions 1 and 2. In either of the two cases the two functions $f_J, f_{\bar{J}}$ only take finitely many values. For the sake of contradiction, assume one of the functions takes two distinct values a_1, a_2 , each with probability greater than zero. W.l.o.g. we assume this is f_J . We denote the preimages of a_1 and a_2 with $A_1, A_2 \subseteq \mathbb{R}^J$. Furthermore, let b be a value that $f_{\bar{J}}$ takes with probability greater than zero and $B \subseteq \mathbb{R}^{\bar{J}}$ its preimage. W.l.o.g. we assume $b \neq a_1$, otherwise, we may switch a_1 and a_2 . In case of assumption 1 we know that $P(A_1 \times B) > 0$.

Under assumption 2 $P(A_1 \times B) > 0$ still holds true as we show in the following. Note that since the density p is strictly positive, a subset of \mathbb{R}^d has some positive probability if and only if it has some positive Lebesgue-measure. This holds true for $A_1 \times \mathbb{R}^{\bar{J}}$ and $\mathbb{R}^J \times B$, because $P(f_J = a_1) > 0$ and $P(f_{\bar{J}} = b) > 0$. Hence, A_1

has some positive J -dimensional Lebesgue-measure and B some positive \bar{J} -dimensional Lebesgue-measure. From that we conclude that $A_1 \times B$ has some positive Lebesgue-measure too and hence $P(A_1 \times B) > 0$ holds true under assumption 2 as well.

But $f_J(A_1 \times B) = a_1 \neq b = f_{\bar{J}}(A_1 \times B)$ and therefore $P(f_J \neq f_{\bar{J}}) > 0$, which is a contradiction. Thus, f_J and $f_{\bar{J}}$ are both constant a.s.

Let us now consider assumption 3. The proof follows a very similar argumentation here. Let us again assume for the sake of contradiction that one of the two functions is not almost surely constant, w.l.o.g. we again assume this to be f_J , and let a_1, a_2 be two values that f_J takes. As before, let b be a value that $f_{\bar{J}}$ takes and assume w.l.o.g. that $b \neq a_1$. If $P(f_J = a_1) > 0$ and $P(f_{\bar{J}} = b) > 0$, the rest of the proof follows exactly as for assumption 2. Otherwise, pick two points $z \in f_J^{-1}(a_1) \subseteq \mathbb{R}^J, w \in f_{\bar{J}}^{-1}(b) \subseteq \mathbb{R}^{\bar{J}}$. Since the functions f_J and $f_{\bar{J}}$ are continuous, points close to z or w map to points close to a_1 or b respectively. More formally, there exist small balls $A_1 := B_{\varepsilon_1}(z) \subseteq \mathbb{R}^J$ around z and $B := B_{\varepsilon_2}(w) \subseteq \mathbb{R}^{\bar{J}}$ around w for some $\varepsilon_1, \varepsilon_2 > 0$, such that $f_J(A_1) \cap f_{\bar{J}}(B) = \emptyset$. By construction, A_1 has some positive J -dimensional Lebesgue-measure and B some positive (\bar{J})-dimensional Lebesgue-measure. Hence, also $A_1 \times B$ has some positive Lebesgue-measure, which again implies $P(A_1 \times B) > 0$. But due to $f_J(A_1) \cap f_{\bar{J}}(B) = \emptyset$ we again receive $P(f_J \neq f_{\bar{J}}) > 0$, which is a contradiction. So, f_J and $f_{\bar{J}}$ are constant a.s. \square

A.1.4 Proof of Theorem 6 and Proposition 2

Proposition 2 is a special case of Theorem 6 resulting from X_J and $X_{\bar{J}}$ being independent and $h^* = 0$.

Proof of Theorem 6. Note that g^* is also the best $\mathcal{L}^2(P)$ -approximation of f^* due to Lemma 11 and so, h^* is the pure interaction of f^* . We begin by simply expanding $v(D) = \text{Var}(f^*)$. We receive

$$\begin{aligned} \text{Var}(f^*) &= \text{Var}(g_J^* + g_{\bar{J}}^* + h^*) \\ &= \text{Var}(g_J^*) + \text{Var}(g_{\bar{J}}^*) + \text{Var}(h^*) \\ &\quad + 2 \text{Cov}(g_J^*, g_{\bar{J}}^*) + \underbrace{2 \text{Cov}(g_J^*, h^*)}_{=0} + \underbrace{2 \text{Cov}(g_{\bar{J}}^*, h^*)}_{=0}, \end{aligned} \tag{1}$$

where the last two summands vanish due to Theorem 5 and Lemma 10. We now consider the approximations based on only one of the two subsets of features. Remember that $E(Y | X_J) = E(f^* | X_J)$ as shown in Lemma 11. We compute

$$\begin{aligned} E(Y | X_J) &= E(f^* | X_J) = E(g_J^* | X_J) + E(g_{\bar{J}}^* | X_J) + \underbrace{E(h^* | X_J)}_{=0} \\ &= g_J^* + E(g_{\bar{J}}^* | X_J), \\ E(Y | X_{\bar{J}}) &= g_{\bar{J}}^* + E(g_J^* | X_{\bar{J}}), \end{aligned}$$

where we again used the fact that h^* is a pure interaction. Computing variances leads to

$$\begin{aligned} v(J) &= \text{Var}(g_J^*) + \text{Var}(E(g_{\bar{J}}^* | X_J)) + 2 \text{Cov}(g_J^*, E(g_{\bar{J}}^* | X_J)) \\ v(\bar{J}) &= \text{Var}(g_{\bar{J}}^*) + \text{Var}(E(g_J^* | X_{\bar{J}})) + 2 \text{Cov}(g_{\bar{J}}^*, E(g_J^* | X_{\bar{J}})). \end{aligned}$$

Using the law of total covariance we can simplify

$$\begin{aligned} \text{Cov}(g_J^*, E(g_{\bar{J}}^* | X_J)) &= \text{Cov}(E(g_J^* | X_J), E(g_{\bar{J}}^* | X_J)) \\ &= \text{Cov}(g_J^*, g_{\bar{J}}^*) - \underbrace{E(\text{Cov}(g_J^*, g_{\bar{J}}^* | X_J))}_{=0} \end{aligned}$$

and analogously,

$$\text{Cov}(g_{\bar{J}}^*, E(g_J^* | X_{\bar{J}})) = \text{Cov}(g_J^*, g_{\bar{J}}^*),$$

leading to

$$v(J) = \text{Var}(g_J^*) + \text{Var}(E(g_{\bar{J}}^* | X_J)) + 2 \text{Cov}(g_J^*, g_{\bar{J}}^*) \tag{2}$$

$$v(\bar{J}) = \text{Var}(g_{\bar{J}}^*) + \text{Var}(E(g_J^* | X_{\bar{J}})) + 2 \text{Cov}(g_{\bar{J}}^*, g_J^*). \tag{3}$$

Putting the equations (1), (2) and (3) together proves the claim. \square

A.2 Examples

A.2.1 Example 3: Linear Function on Normally Distributed Data

We consider

$$Y = X_1 + X_2 + cX_1X_2$$

where $X \sim \mathcal{N}\left(0, \begin{pmatrix} 1 & \beta \\ \beta & 1 \end{pmatrix}\right)$ for arbitrary $c \in \mathbb{R}, \beta \in [0, 1)$. In the following, we will derive the value functions, the functional decomposition and the DIP decomposition in a purely theoretical way.

Mathematical Basics and Precomputations For all these computations we need the generalized Isserlis theorem, formalized in Withers (1985).

Theorem (Withers, 1985). *If (X_1, \dots, X_d) is a zero-mean multivariate normal random vector and $A = \{\alpha_1, \dots, \alpha_l\}$ is a subset of (not necessarily distinct) indices between 1 and d , we have*

$$\mathbb{E}(X_{\alpha_1} \cdots X_{\alpha_l}) = \sum_{p \in P_A^2} \prod_{\{i,j\} \in p} \mathbb{E}(X_i X_j).$$

Here, P_A^2 is the set of all possible partitions of the set A into pairs. The product then goes over all these pairs in a particular partition p .

This means in particular that the expression vanishes if l is odd, because in this case there are zero possible partitions of $\{\alpha_1, \dots, \alpha_l\}$ into pairs.

With the aid of this theorem we compute the following expressions that will be used afterwards. Note that $\mathbb{E}(X_1^2) = \mathbb{E}(X_2^2) = 1$ and $\mathbb{E}(X_1 X_2) = \beta$.

$$\begin{aligned} \text{Var}(X_1 X_2) &= \mathbb{E}(X_1^2 X_2^2) - \mathbb{E}(X_1 X_2)^2 \\ &= \mathbb{E}(X_1^2) \mathbb{E}(X_2^2) + \mathbb{E}(X_1 X_2)^2 + \mathbb{E}(X_1 X_2)^2 - \mathbb{E}(X_1 X_2)^2 \\ &= \mathbb{E}(X_1^2) \mathbb{E}(X_2^2) + \mathbb{E}(X_1 X_2)^2 \\ &= 1 + \beta^2, \\ \text{Var}(X_1^2) &= \mathbb{E}(X_1^4) - \mathbb{E}(X_1^2)^2 = 3 \mathbb{E}(X_1^2)^2 - \mathbb{E}(X_1^2)^2 = 2 \mathbb{E}(X_1^2)^2 \\ &= 2, \\ \text{Cov}(X_1, X_1 X_2) &= \mathbb{E}(X_1^2 X_2) - \mathbb{E}(X_1) \mathbb{E}(X_1 X_2) = 0, \\ \text{Cov}(X_2, X_1 X_2) &= 0, \\ \text{Cov}(X_1, X_1^2) &= \mathbb{E}(X_1^3) - \mathbb{E}(X_1^2) \mathbb{E}(X_1) = 0, \\ \text{Cov}(X_2, X_2^2) &= 0 \\ \text{Cov}(X_1 X_2, X_1^2) &= \mathbb{E}(X_1^3 X_2) - \mathbb{E}(X_1 X_2) \mathbb{E}(X_1^2) = 3 \mathbb{E}(X_1^2) \mathbb{E}(X_1 X_2) - \mathbb{E}(X_1 X_2) \mathbb{E}(X_1^2) \\ &= 3\beta - \beta = 2\beta \\ \text{Cov}(X_1 X_2, X_2^2) &= 2\beta \\ \text{Cov}(X_1^2, X_2^2) &= \mathbb{E}(X_1^2 X_2^2) - \mathbb{E}(X_1^2) \mathbb{E}(X_2^2) \\ &= \mathbb{E}(X_1^2) \mathbb{E}(X_2^2) + 2 \mathbb{E}(X_1 X_2)^2 - \mathbb{E}(X_1^2) \mathbb{E}(X_2^2) = 2 \mathbb{E}(X_1 X_2)^2 \\ &= 2\beta^2. \end{aligned}$$

Computation of the Value Functions With these expressions, we can now compute

$$\begin{aligned} v(1 \cup 2) &= \text{Var}(Y) = \text{Var}(X_1 + X_2 + cX_1X_2) \\ &= \text{Var}(X_1) + \text{Var}(X_2) + c^2 \text{Var}(X_1X_2) + 2 \text{Cov}(X_1, X_2) + \underbrace{2c \text{Cov}(X_1, X_1X_2)}_{=0} + \underbrace{2c \text{Cov}(X_2, X_1X_2)}_{=0} \\ &= 2 + 2\beta + c^2(1 + \beta^2). \end{aligned}$$

Note that $E(X_2 | X_1) = \beta X_1$, which is known from the conditional distribution of a multivariate normal. With this identity in mind, we compute

$$\begin{aligned}
 v(1) &= \text{Var}(E(Y | X_1)) = \text{Var}(E(X_1 + X_2 + cX_1X_2 | X_1)) \\
 &= \text{Var}(X_1 + E(X_2 | X_1) + cX_1 E(X_2 | X_1)) = \text{Var}(X_1 + \beta X_1 + c\beta X_1^2) \\
 &= (1 + \beta)^2 \text{Var}(X_1) + c^2 \beta^2 \text{Var}(X_1^2) + 2(1 + \beta)c\beta \underbrace{\text{Cov}(X_1, X_1^2)}_{=0} \\
 &= (1 + \beta)^2 + 2c^2 \beta^2.
 \end{aligned}$$

Analogously,

$$v(2) = (1 + \beta)^2 + 2c^2 \beta^2.$$

Plugging in $c = \beta = 0$ and $c = \sqrt{6}, \beta = 0.5$ from our two DGPs in Example 3 and dividing by $\text{Var}(Y) = 2 + 2\beta + c^2(1 + \beta^2)$ we receive

$$\bar{v}(1 \cup 2) = 1, \quad \bar{v}(1) = \frac{1}{2}, \quad \bar{v}(2) = \frac{1}{2}$$

as claimed.

Functional Decomposition For the components of the cooperative impact, we first need the functions g_1^*, g_2^* and h^* . Note that $Y = f^*(X)$. The component functions are given by

$$\begin{aligned}
 g_1^*(X_1) &= X_1 + \frac{c\beta}{1 + \beta^2} X_1^2 - \frac{c\beta(1 - \beta^2)}{2(1 + \beta^2)} \\
 g_2^*(X_2) &= X_2 + \frac{c\beta}{1 + \beta^2} X_2^2 - \frac{c\beta(1 - \beta^2)}{2(1 + \beta^2)} \\
 h^*(X) &= cX_1X_2 - \frac{c\beta}{1 + \beta^2} X_1^2 - \frac{c\beta}{1 + \beta^2} X_2^2 + \frac{c\beta(1 - \beta^2)}{1 + \beta^2}.
 \end{aligned}$$

To prove this, we verify $E(h^* | X_1) = E(h^* | X_2) = 0$, which is equivalent to g^* being the $\mathcal{L}^2(P)$ -optimal GAM as we showed in Theorem 5. We first note that

$$E(X_2^2 | X_1) = \text{Var}(X_2 | X_1) + E(X_2 | X_1)^2 = 1 - \beta^2 + \beta^2 X_1^2$$

where $\text{Var}(X_2 | X_1)$ and $E(X_2 | X_1)$ are known from the conditional distributions of the multivariate normal distribution. We then compute

$$\begin{aligned}
 E(h^* | X_1) &= cX_1 E(X_2 | X_1) - \frac{c\beta}{1 + \beta^2} X_1^2 - \frac{c\beta}{1 + \beta^2} E(X_2^2 | X_1) + \frac{c\beta(1 - \beta^2)}{1 + \beta^2} \\
 &= c\beta X_1^2 - \frac{c\beta}{1 + \beta^2} X_1^2 - \frac{c\beta(1 - \beta^2)}{1 + \beta^2} - \frac{c\beta^3}{1 + \beta^2} X_1^2 + \frac{c\beta(1 - \beta^2)}{1 + \beta^2} \\
 &= c\beta X_1^2 - \frac{c\beta}{1 + \beta^2} X_1^2 - \frac{c\beta^3}{1 + \beta^2} X_1^2 \\
 &= \frac{c\beta(1 + \beta^2) - c\beta}{1 + \beta^2} X_1^2 - \frac{c\beta^3}{1 + \beta^2} X_1^2 \\
 &= \frac{c\beta^3}{1 + \beta^2} X_1^2 - \frac{c\beta^3}{1 + \beta^2} X_1^2 \\
 &= 0.
 \end{aligned}$$

Analogously, one can compute $E(h^* | X_2) = 0$. Hence, $Y - h^*(X)$ is the $\mathcal{L}^2(P)$ -optimal GAM for Y . The two components g_1^* and g_2^* are unique up to a constant as we know from Theorem 12. We simply split the additive constant equally between the two. For the computation of the variance they can simply be dropped.

Decomposition of the Cooperative Impact Again taking use of the quantities we computed with the generalized Isserlis theorem, we can now determine the components of the cooperative impact. For the cross-predictability we get

$$\begin{aligned}
 \text{Var}(\text{E}(g_1^* | X_2)) &= \text{Var}\left(\text{E}(X_1 | X_2) + \frac{c\beta}{1+\beta^2} \text{E}(X_1^2 | X_2)\right) \\
 &= \text{Var}\left(\beta X_2 + \frac{c\beta}{1+\beta^2} (1 + \beta^2 + \beta^2 X_2^2)\right) \\
 &= \text{Var}\left(\beta X_2 + \frac{c\beta}{1+\beta^2} (\beta^2 X_2^2)\right) \\
 &= \beta^2 \text{Var}(X_2) + \left(\frac{c\beta^3}{1+\beta^2}\right)^2 \text{Var}(X_2^2) + 2 \frac{c\beta^4}{1+\beta^2} \underbrace{\text{Cov}(X_2, X_2^2)}_{=0} \\
 &= \beta^2 + \frac{2c^2\beta^6}{(1+\beta^2)^2}
 \end{aligned}$$

and in the same manner

$$\text{Var}(\text{E}(g_2^* | X_1)) = \beta^2 + \frac{2c^2\beta^6}{(1+\beta^2)^2}.$$

Summed up we have

$$\text{Var}(\text{E}(g_1^* | X_2)) + \text{Var}(\text{E}(g_2^* | X_1)) = 2\beta^2 + \frac{4c^2\beta^6}{(1+\beta^2)^2}.$$

For the covariance we compute

$$\begin{aligned}
 \text{Cov}(g_1^*, g_2^*) &= \text{Cov}\left(X_1 + \frac{c\beta}{1+\beta^2} X_1^2, X_2 + \frac{c\beta}{1+\beta^2} X_2^2\right) \\
 &= \text{Cov}(X_1, X_2) + \frac{c\beta}{1+\beta^2} \underbrace{\text{Cov}(X_1, X_2^2)}_{=0} \\
 &\quad + \frac{c\beta}{1+\beta^2} \underbrace{\text{Cov}(X_2, X_1^2)}_{=0} + \left(\frac{c\beta}{1+\beta^2}\right)^2 \text{Cov}(X_1^2, X_2^2) \\
 &= \beta + \frac{2c^2\beta^4}{(1+\beta^2)^2}.
 \end{aligned}$$

Eventually, for the interaction surplus we have

$$\begin{aligned}
 \text{Var}(h^*) &= \text{Var}\left(cX_1X_2 - \frac{c\beta}{1+\beta^2} X_1^2 - \frac{c\beta}{1+\beta^2} X_2^2\right) \\
 &= c^2 \text{Var}(X_1X_2) + \frac{c^2\beta^2}{(1+\beta^2)^2} \text{Var}(X_1^2) + \frac{c^2\beta^2}{(1+\beta^2)^2} \text{Var}(X_2^2) \\
 &\quad - 2 \frac{c^2\beta}{1+\beta^2} \text{Cov}(X_1X_2, X_1^2) - 2 \frac{c^2\beta}{1+\beta^2} \text{Cov}(X_1X_2, X_2^2) \\
 &\quad + 2 \frac{c^2\beta^2}{(1+\beta^2)^2} \text{Cov}(X_1^2, X_2^2) \\
 &= c^2 (1 + \beta^2) + 4 \frac{c^2\beta^2}{(1+\beta^2)^2} - 8 \frac{c^2\beta^2}{1+\beta^2} + 4 \frac{c^2\beta^4}{(1+\beta^2)^2} \\
 &= c^2 (1 + \beta^2) + 4 \frac{c^2\beta^2 (1 + \beta^2)}{(1+\beta^2)^2} - 8 \frac{c^2\beta^2}{1+\beta^2} \\
 &= c^2 (1 + \beta^2) - 4 \frac{c^2\beta^2}{1+\beta^2}.
 \end{aligned}$$

Plugging in the values for c and β delivers the values of the DIP decomposition. Those can be normalized by dividing by $\text{Var}(Y) = 2 + 2\beta + c^2(1 + \beta^2)$ to receive the values presented in the paper.

A.2.2 Main Effect Dependencies Cannot Be Derived by Only Value Functions and Feature Correlation

In the following, we give an example of three two-dimensional DGPs, where X follows the same distribution in each of the three and the value functions for all subsets of features coincide. However, we get different DIP decompositions for each DGP, illustrating that we cannot deduce the main effect dependencies from just the dependencies of the features, even if all the value functions are known.

Consider the following DGPs:

DGP 1:

$$X \sim \mathcal{N}\left(0, \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}\right)$$

$$Y = -4.3X_1 - 0.9X_2 - 3.9X_1^2 + 3.0X_2^2$$

DGP 2:

$$X \sim \mathcal{N}\left(0, \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}\right)$$

$$Y = -1.3X_1 - 4.7X_2 + 3.6X_1^2 - 3.0X_2^2 + 4.7X_1X_2$$

DGP 3:

$$X \sim \mathcal{N}\left(0, \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}\right)$$

$$Y = 10.9X_1 + 2.4X_2 - 5.1X_1^2 - 5.3X_2^2 + 11.3X_1X_2.$$

In each of the three cases, we get the (empirical and rounded) normalized value functions

$$\bar{v}(1) = 0.7, \quad \bar{v}(2) = 0.3, \quad \bar{v}(1 \cup 2) = 1.$$

Despite the coinciding value functions and the same correlation, we get three different DIP decompositions as the following plot shows.

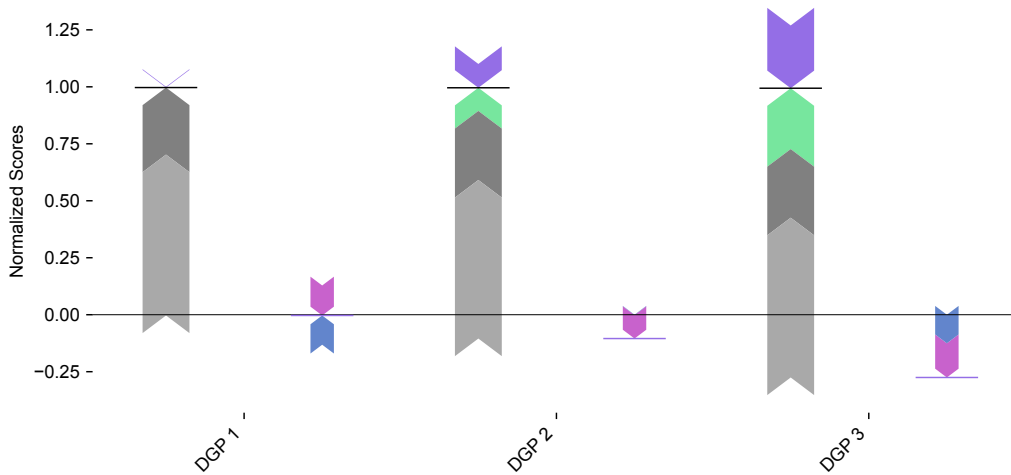


Figure 4: For each example, we show a forceplot visualizing the DIP decomposition into standalone contributions ($v(1)$ and $v(2)$), **main effect dependencies** ($\text{Dep}(1, 2)$) and **interaction surplus**, where the direction of each bar (upward or downward) represents the sign. They sum up to $v(1, 2)$ (black horizontal line). The slim bars (right) show the decomposition of $\text{Dep}(1, 2)$ (purple horizontal line) into **covariance** and **cross-predictability**.

A.2.3 Vanishing Main Effect Dependencies Despite Variable Dependence

In the following, we show an example of a two-dimensional DGP, where the two features are strongly correlated, but their main effect dependencies vanish.

Let $Z_0, Z_1, Z_2 \sim \text{Unif}(0, \dots, 9)$ be independently distributed. We define

$$\begin{aligned} X_1 &= 10Z_0 + Z_1 \\ X_2 &= 10Z_0 + Z_2 \\ Y &= (X_1 \bmod 10) + (X_2 \bmod 10) = Z_1 + Z_2, \end{aligned}$$

that is, Z_0 defines the first digit and Z_1, Z_2 define the second digit of X_1 and X_2 . The target variable Y is just the sum of their last digits.

Clearly, X_1 and X_2 are strongly correlated due to Z_0 . However, we have

$$\begin{aligned} g_1(X_1) &= X_1 \bmod 10 = Z_1 \\ g_2(X_2) &= X_2 \bmod 10 = Z_2 \end{aligned}$$

and therefore $E(g_1 | X_2) = E(Z_1 | 10Z_0 + Z_2) = E(Z_1)$, $E(g_2 | X_1) = E(Z_2 | 10Z_0 + Z_1) = E(Z_2)$. Hence, the cross-predictability

$$\text{Var}(E(g_1 | X_2)) + \text{Var}(E(g_2 | X_1)) = 0$$

as well as the main effect covariance

$$2\text{Cov}(g_1, g_2) = \text{Cov}(Z_1, Z_2) = 0$$

vanish.

Here, we see an example where all of the dependencies between X_1 and X_2 stem from their shared first digit Z_0 , which is independent from the target Y . All of their information about Y is contained in their last digits Z_1, Z_2 , which are independent. This means that although our variables have a strong dependencies, they do not share information about the target, so their main effects dependencies vanish. This once more illustrates that main order dependencies cannot be determined by just the correlation, rather, they measure the dependencies of only those parts of our features that are relevant for predicting.

A.2.4 Binary Examples 7 - 9

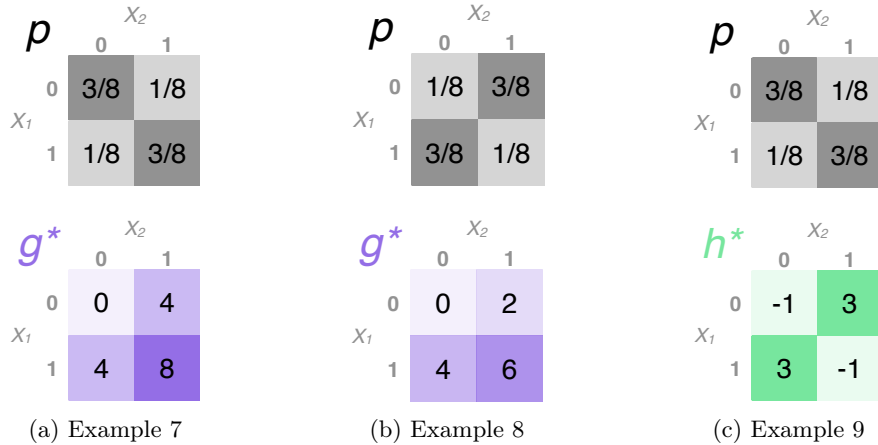


Figure 5: Visualizing the data generating process of the three illustrative student examples.

Example 7 Since $Y = 4X_1 + 4X_2$ can exactly be determined by a GAM, we have $f^*(X) = g^*(X) = 4X_1 + 4X_2$ and $h^* = 0$. Note that

$$\begin{aligned} E(X_2 | X_1 = 1) &= \frac{3}{4}, \\ E(X_2 | X_1 = 0) &= \frac{1}{4}. \end{aligned}$$

We can also write this as $E(X_2 | X_1) = 0.5X_1 + 0.25$. In the same way, $E(X_1 | X_2) = 0.5X_2 + 0.25$. The best univariate predictors are given by

$$\begin{aligned} E(Y | X_1) &= E(4X_1 + 4X_2 | X_1) = 4X_1 + 4(0.5X_1 + 0.25) = 6X_1 + 1, \\ E(Y | X_2) &= 6X_2 + 1. \end{aligned}$$

The variance of a $\text{Ber}(p)$ -distributed variable is given by $p(1-p)$, so $\text{Var}(X_i) = 0.25$ in our case. This leads to

$$\begin{aligned} v(1) &= \text{Var}(E(Y | X_1)) = \text{Var}(6X_1 + 1) = 36 \text{Var}(X_1) = 9 \\ v(2) &= 9 \\ v(1 \cup 2) &= \text{Var}(Y) = E(Y^2) - E(Y)^2 = \frac{3}{8} \cdot 8^2 + 2 \cdot \frac{1}{8} \cdot 4^2 - 4^2 = 12. \end{aligned}$$

This leads to main effect dependencies of 6 because there is no interaction surplus. For main effect covariance we get

$$2 \text{Cov}(4X_1, 4X_2) = 32 \text{Cov}(X_1, X_2) = 32(E(X_1X_2) - E(X_1)E(X_2)) = 32 \cdot \left(\frac{3}{8} - \frac{1}{4}\right) = 32 \cdot \frac{1}{8} = 4,$$

leading to a cross-predictability of $6 - 4 = 2$.

Example 8 Again, $f^*(X) = g^*(X) = 4X_1 + 2X_2$ and $h^* = 0$. This time,

$$\begin{aligned} E(X_2 | X_1 = 1) &= \frac{1}{4}, \\ E(X_2 | X_1 = 0) &= \frac{3}{4}, \end{aligned}$$

which means $E(X_2 | X_1) = -0.5X_1 + 0.25$ and $E(X_1 | X_2) = -0.5X_2 + 0.25$. For the univariate predictors we compute

$$\begin{aligned} E(Y | X_1) &= E(4X_1 + 2X_2 | X_1) = 4X_1 + 2(-0.5X_1 + 0.25) = 3X_1 + 0.5, \\ E(Y | X_2) &= E(4X_1 + 2X_2 | X_2) = 4 \cdot (-0.5X_2 + 0.25) + 2X_2 = 1, \end{aligned}$$

leading to

$$\begin{aligned} v(1) &= \text{Var}(E(Y | X_1)) = \text{Var}(3X_1 + 0.5) = 9 \text{Var}(X_1) = 2.25 \\ v(2) &= \text{Var}(1) = 0 \\ v(1 \cup 2) &= \text{Var}(Y) = E(Y^2) - E(Y)^2 = \frac{1}{8} \cdot 8^2 + \frac{3}{8} \cdot 4^2 + \frac{3}{8} \cdot 2^2 - 3^2 = 3. \end{aligned}$$

This implies $\text{Dep}(1, 2) = -0.75$ because again, the interaction surplus vanishes. The main effect covariance is given by

$$2 \text{Cov}(4X_1, 2X_2) = 16 \text{Cov}(X_1, X_2) = 16(E(X_1X_2) - E(X_1)E(X_2)) = 16 \cdot \left(\frac{1}{8} - \frac{1}{4}\right) = -2,$$

so the cross-predictability is $-0.75 + 2 = 1.25$.

Example 9 Note that the distribution of X is the same as in Example 7. We first prove that the decomposition $Y = g^*(X) + h^*(X)$ for $g^*(X) = 4X_1 + 4X_2$ and $h^*(X) = 4(X_1 \oplus X_2) - 1$ is indeed the unique functional decomposition into a GAM and a pure interaction. According to Theorem 5 it is sufficient to prove $E(h^* | X_1) = E(h^* | X_2) = 0$. We get

$$\begin{aligned} E(4(X_1 \oplus X_2) - 1 | X_1 = 1) &= \underbrace{P(X_2 = 1 | X_1 = 1)}_{=\frac{3}{4}} \cdot (0 - 1) + \underbrace{P(X_2 = 0 | X_1 = 1)}_{=\frac{1}{4}} \cdot (4 - 1) = 0 \\ E(4(X_1 \oplus X_2) - 1 | X_1 = 0) &= \underbrace{P(X_2 = 1 | X_1 = 0)}_{=\frac{1}{4}} \cdot (4 - 1) + \underbrace{P(X_2 = 0 | X_1 = 0)}_{=\frac{3}{4}} \cdot (0 - 1) = 0. \end{aligned}$$

So, it holds $E(h^* | X_1) = 0$ and due to the symmetry also $E(h^* | X_2) = 0$. In particular, $E(h^*) = E(E(h^* | X_1)) = 0$, so h^* is centered. The univariate predictor yields

$$E(Y | X_1) = E(g^* | X_1) + \underbrace{E(h^* | X_1)}_{=0} = E(4X_1 + 4X_2 | X_1),$$

likewise for X_2 . This means, the univariate predictors are the same as in Example 7, from which we may conclude that also $v(1) = v(2) = 9$. Since g^* as well coincides with the one in Example 7, we can also conclude that $\text{Dep}(1, 2) = 6$, the cross-predictability equals 2 and the main effects covariance equals 4. For the interaction surplus we compute

$$\text{Var}(h^*) = E(h^2) - \underbrace{E(h)^2}_{=0} = \frac{3}{4} \cdot (-1)^2 + \frac{1}{4} \cdot 3^2 = 3.$$

From this, we right away conclude

$$v(1 \cup 2) = v(1) + v(2) + \text{Var}(h^*) - \text{Dep}(1, 2) = 15.$$

A.3 Decomposition of SAGE

The SAGE value (Covert et al., 2020), also known as Shapley effect (Song et al., 2016) of a feature j is defined as

$$\Phi_j := \sum_{S \subseteq D \setminus j} c_S \cdot (v(S \cup j) - v(S)), \quad (4)$$

where the weights c_S are given by

$$c_S = \frac{(d - |S| - 1)!|S|!}{d!}.$$

Note that $\sum_{S \subseteq D \setminus j} c_S = 1$. In the following, we write $\text{Int}(j, S)$ for the interaction surplus and $\text{Dep}(j, S)$ for the main order dependencies between the feature groups j and S . By decomposing every summand of (4) we receive

$$\begin{aligned} \Phi_j &:= \sum_{S \subseteq D \setminus j} c_S \cdot (v(S \cup j) - v(S)) = \sum_{S \subseteq D \setminus j} c_S \cdot (v(j) + \text{Int}(j, S) - \text{Dep}(j, S)) \\ &= v(j) + \sum_{S \subseteq D \setminus j} c_S \cdot \text{Int}(j, S) - \sum_{S \subseteq D \setminus j} c_S \cdot \text{Dep}(j, S), \end{aligned}$$

yielding a decomposition of Φ_j into its standalone contribution, the contribution stemming from interactions and the one stemming from dependencies. Here, instead of only computing the interaction surplus and the main order dependencies between j and \bar{j} , we do so for j and every subset $S \subseteq D \setminus j$ and compute a weighted average, just as in the formula for the Shapley effect.

One disadvantage of the Shapley effect over LOCO is the exponential runtime in the number of features since we need to refit a model for every $S \subseteq D \setminus j$ or at least use a sufficiently large number of subsets for an approximation.

B DETAILS ON ESTIMATION AND IMPLEMENTATION

B.1 Estimation

In Theorem 6 we derived a decomposition of the cooperative impact, assuming the optimal models f^* and g^* and the true data distribution $(X, Y) \sim P$ to be known. In practice, the optimal models are not available; Instead we fit and evaluate ML models using some dataset $((x, y)^{(1)}, \dots, (x, y)^{(n)})$ with indices $\mathcal{I} = \{1, \dots, n\}$. To avoid bias due to overfitting, we reformulate the quantities such that they can be estimated using test set performance. We start by reformulating interaction surplus and main effect dependencies in terms of value functions.

Interaction Surplus And Main Effect Dependencies In Terms of Value Functions First, we recall that we defined the interaction surplus for optimal model f^* and optimal GGAM g^* as $\text{Var}(h^*)$ where $h^* := f^* - g^*$. In Lemma 13, we show that we can equivalently define the interaction surplus, in the following denoted as $\text{Int}(J, \bar{J})$, as the difference in value function between the full model f^* and the GGAM g^* .

Lemma 13. *Let $(X, Y) \sim P$ be a DGP, $J \subseteq D$ a subset of features, f^* the $\mathcal{L}^2(P)$ -optimal predictor and $g^* = g_J^* + g_{\bar{J}}^*$ the $\mathcal{L}^2(P)$ -optimal GGAM in X_J and $X_{\bar{J}}$. We call $h^* = f^* - g^*$. Then, the interaction surplus $\text{Int}(J, \bar{J}) := \text{Var}(h^*)$ is given by*

$$\text{Int}(J, \bar{J}) = v_{f^*}(J \cup \bar{J}) - v_{g^*}(J \cup \bar{J}).$$

Proof. We compute

$$\begin{aligned} v_{f^*}(J \cup \bar{J}) - v_{g^*}(J \cup \bar{J}) &= \mathbb{E}((Y - g^*)^2) - \mathbb{E}((Y - f^*)^2) \\ &= \mathbb{E}((Y - f^* + h^*)^2) - \mathbb{E}((Y - f^*)^2) \\ &= \mathbb{E}((Y - f^*)^2) + \underbrace{2\mathbb{E}((Y - f^*) \cdot h^*)}_{=0} + \mathbb{E}((h^*)^2) - \mathbb{E}((Y - f^*)^2) \\ &= \text{Var}(h^*). \end{aligned}$$

Here, $\mathbb{E}((Y - f^*) \cdot h^*)$ vanishes because f^* is the optimal predictor, and its residual is hence perpendicular to every function in $\mathcal{L}^2(\mathbb{R}^d, P)$ (Luenberger, 1997). Furthermore, h^* is mean-centered due to Theorem 5 and thus $\mathbb{E}(h^2) = \text{Var}(h^2)$. \square

Estimation of Dep and Int As a consequence of Lemma 13, we can estimate the interaction surplus for some \hat{f} , \hat{g} and observation indices \mathcal{I} as the difference in estimated value functions. Therefore, we first recapitulate that the definition the value function for the \mathcal{L}^2 -loss and a model f is given by

$$v_{f, \mathcal{L}^2}(S) := \mathbb{E}((f_\emptyset - Y)^2) - \mathbb{E}((f_S(X_S) - Y)^2),$$

which can be estimated in terms of test data with indices \mathcal{I}_{te} using the empirical risk by

$$\hat{v}_{f, \mathcal{L}^2}^{\mathcal{I}_{te}}(S) = \frac{1}{|\mathcal{I}_{te}|} \left(\sum_{i \in \mathcal{I}_{te}} (f_\emptyset - y^{(i)})^2 - \sum_{i \in \mathcal{I}_{te}} (f_S(x^{(i)}) - y^{(i)})^2 \right).$$

We recall that f_S can be obtained using refitting or via conditional integration. The term f_\emptyset is the best constant approximation of f , which is $\mathbb{E}(f)$. For an optimal predictor, this is the same as $\mathbb{E}(Y)$. Both of these terms can be approximated by the empirical mean. To avoid biased results, all models must be fit on training data with indices \mathcal{I}_{tr} where $\mathcal{I}_{tr} \cap \mathcal{I}_{te} = \emptyset$.

Based on the empirical value function, we can estimate the interaction surplus for some test set \mathcal{I}_{te} as

$$\widehat{\text{Int}}_{\hat{f}, \hat{g}}^{\mathcal{I}_{te}}(J, \bar{J}) := \hat{v}_{\hat{f}, \mathcal{L}^2}^{\mathcal{I}_{te}}(J \cup \bar{J}) - \hat{v}_{\hat{g}, \mathcal{L}^2}^{\mathcal{I}_{te}}(J \cup \bar{J}).$$

To estimate $\text{Dep}(J, \bar{J})$, we recall that $\Psi(J, \bar{J}) = \text{Var}(h^*) - \text{Dep}(J, \bar{J})$, and thus $\text{Dep}(J, \bar{J}) := \Psi(J, \bar{J}) - \text{Int}(J, \bar{J})$. We can estimate $\Psi(J, \bar{J})$ as

$$\widehat{\Psi}_{\hat{f}}^{\mathcal{I}_{te}}(J, \bar{J}) = \hat{v}_{\hat{f}, \mathcal{L}^2}^{\mathcal{I}_{te}}(J \cup \bar{J}) - \hat{v}_{\hat{f}, \mathcal{L}^2}^{\mathcal{I}_{te}}(J) - \hat{v}_{\hat{f}, \mathcal{L}^2}^{\mathcal{I}_{tr}}(\bar{J})$$

and get

$$\widehat{\text{Dep}}_{\hat{f}, \hat{g}}^{\mathcal{I}^{te}}(J, \bar{J}) := \widehat{\Psi}_{\hat{f}}^{\mathcal{I}^{te}}(J, \bar{J}) - \widehat{\text{Int}}_{\hat{f}, \hat{g}}^{\mathcal{I}^{te}}(J, \bar{J}).$$

We summarize the estimation procedure in Algorithm 1.

Algorithm 1 Estimation of **Int** and **Dep** Using Empirical Risk on Test dData

Input: Feature indices J , data $(x, y)^{\mathcal{I}}$, split into train and test $\mathcal{I}^{tr} \cup \mathcal{I}^{te} = \mathcal{I}$, model \hat{f} .

Output: Estimates of **interaction surplus** $\text{Int}(J, \bar{J})$ and **main effect dependencies** $\text{Dep}(J, \bar{J})$.

- 1: $\hat{f}_J, \hat{f}_{\bar{J}}, \hat{f}_{\emptyset} \leftarrow \mathcal{L}^2$ fits on $(x_S, y)^{\mathcal{I}^{tr}}$ for $S = J, S = \bar{J}$, and $S = \emptyset$.
 - 2: $\hat{g} \leftarrow \mathcal{L}^2$ fit of GGAM in J, \bar{J} on $(x, y)^{\mathcal{I}^{tr}}$
 - 3: $\widehat{\text{Int}}_{\hat{f}, \hat{g}}^{\mathcal{I}^{te}}(J, \bar{J}) \leftarrow \hat{v}_{\hat{f}, \mathcal{L}^2}^{\mathcal{I}^{te}}(J, \bar{J}) - \hat{v}_{\hat{g}, \mathcal{L}^2}^{\mathcal{I}^{te}}(J, \bar{J})$
 - 4: $\widehat{\Psi}_{\hat{f}}^{\mathcal{I}^{te}}(J, \bar{J}) \leftarrow \hat{v}_{\hat{f}, \mathcal{L}^2}^{\mathcal{I}^{te}}(J, \bar{J}) - \hat{v}_{\hat{f}, \mathcal{L}^2}^{\mathcal{I}^{te}}(J) - \hat{v}_{\hat{f}, \mathcal{L}^2}^{\mathcal{I}^{te}}(\bar{J})$ \triangleright uses $\hat{f}_J, \hat{f}_{\bar{J}}, \hat{f}_{\emptyset}$
 - 5: $\widehat{\text{Dep}}_{\hat{f}, \hat{g}}^{\mathcal{I}^{te}}(J, \bar{J}) \leftarrow \widehat{\Psi}_{\hat{f}}^{\mathcal{I}^{te}}(J, \bar{J}) - \widehat{\text{Int}}_{\hat{f}, \hat{g}}^{\mathcal{I}^{te}}(J, \bar{J})$
-

Estimation of Main Effect Cross-Predictability and Covariance We recall that $\text{Dep}(J, \bar{J})$ is the sum of cross-predictability and main effect covariance

$$\text{Dep}(J, \bar{J}) := \underbrace{\text{Var}(\mathbb{E}(g_J^* | X_J)) + \text{Var}(\mathbb{E}(g_{\bar{J}}^* | X_{\bar{J}}))}_{\text{Cross-Predictability CP}(J, \bar{J})} + \underbrace{2 \text{Cov}(g_J^*, g_{\bar{J}}^*)}_{\text{Covariance CO}(J, \bar{J})}.$$

Given an estimate of $\text{Dep}(J, \bar{J})$, we can now either estimate the cross-predictability $\text{CP}(J, \bar{J})$ or the covariance $\text{CO}(J, \bar{J})$ and get the respective other term as the difference of both.

In our experiments, we estimate the covariance of the GGAM components, which is efficient to compute, since the GGAM components are readily available, and the covariance is a comparatively cheap computation (Algorithm 2). On the other hand, estimating the cross-predictability is also possible but requires two further fits for approximating $\mathbb{E}(\hat{g}_J | X_J)$ and $\mathbb{E}(\hat{g}_{\bar{J}} | X_{\bar{J}})$.

Algorithm 2 Estimation of **cross-predictability** and **covariance**

Input: $\widehat{\text{Dep}}_{\hat{f}, \hat{g}}^{\mathcal{I}^{te}}(J, \bar{J})$, GGAM $\hat{g} = \hat{g}_J + \hat{g}_{\bar{J}}$ fitted on $(x, y)^{\mathcal{I}^{tr}}$ to minimize \mathcal{L}^2

Output: Estimates of **cross-predictability** $\text{CP}(J, \bar{J})$ and **covariance** $\text{CO}(J, \bar{J})$

- 1: $\widehat{\text{CO}}_{\hat{f}, \hat{g}}^{\mathcal{I}^{te}}(J, \bar{J}) \leftarrow 2 \text{Cov}(\hat{g}_J(x^{\mathcal{I}^{te}}), \hat{g}_{\bar{J}}(x^{\mathcal{I}^{tr}}))$
 - 2: $\widehat{\text{CP}}_{\hat{f}, \hat{g}}^{\mathcal{I}^{te}}(J, \bar{J}) \leftarrow \widehat{\text{Dep}}_{\hat{f}, \hat{g}}^{\mathcal{I}^{te}}(J, \bar{J}) - \widehat{\text{CO}}_{\hat{f}, \hat{g}}^{\mathcal{I}^{te}}(J, \bar{J})$
-

B.2 Implementation and Code

Python Package We implemented the method in a python package called `dipd`. The package and installation instructions are available via <https://github.com/gcskoenig/dipd>. The package is publicly available on GitHub and pypi.

For the GGAMs, we rely on the `interpretML` package that implements so-called explainable boosting machines (Nori et al., 2019). Furthermore, we use `numpy`, `pandas`, `matplotlib`, `seaborn`, `tqdm`, `scipy`, `statsmodels`, and `scikit-learn` in the most current version available in python3.11.7 (a full list of the installed packages including version can be found in the linked repository).

In all experiments involving the explainable boosting machine, we fit the model using the default hyperparameters, except that we specify which interventions are and are not allowed. In cases where we use a linear model as GAM (or GGAM), we use the OLS implementation in the `statsmodels` package, also using default hyperparameters.

Experiments To reproduce our experiments, we refer to the instructions in our repository (https://github.com/gcskoenig/aistats_2025_DIP). As follows, we summarize the key parameters for each of the experiments, that is, which models were used, how much data was used, and how we split test and training data.

For the student examples (Example 7 to 9), we sampled 10^5 observations and randomly split the dataset into 80% training and 20% test data. For the first two examples, we use a linear model as GAM; for the interaction example, the explainable boosting machine.

For Example 3, where the cooperative forces cancel out, we sample 10^5 data points, again hold out 20 % of the data for testing, and leverage explainable boosting machines for all model fits.

For the real-world applications, we split the data into 10 folds, train on the respective training, and compute the scores on the test data. For all model fits, we leverage the explainable boosting machines.

For the example introduced in Appendix A.2.2 we sampled 10^6 observations and used explainable boosting machines for all model fits.

In Appendix C we present additional experimental results. In all cases we use test train splits with 20% test data and rely on explainable boosting machines.

Compute The experiments were run on a MacBook Pro with M3 Pro Chip. The illustrative examples all ran in less than one minute, the DIP decompositions of LOCO on the housing and wine dataset took about ten minutes each. The DIP decompositions of SAGE on the wine and housing datasets (Appendix C.2) took about one and a half hours each. The DIP decompositions of LOCO for the superconductivity/online news datasets (Appendix C.3) took four hours/ten minutes.

C EXPERIMENTAL RESULTS

In the main paper, we decomposed the LOCO scores on the wine quality dataset using DIP (Section 8, Figure 3 left). In this section, we present additional experimental results. In Section C.1, we investigate the pairwise relationships of variables using DIP. In Appendix C.2, we additionally present the results of a DIP decomposition of SAGE values. The code for all presented experiments is available in the repository accompanying the paper (https://github.com/gcskoenig/aistats_2025_DIP).

For more detailed results regarding the experiments in the main paper (such as standard deviations or the results for individual folds) we refer to the repository accompanying the paper.

C.1 Detailed Analysis on the Wine Quality Dataset

C.1.1 Pairwise Decompositions Reveal Cooperation Partners

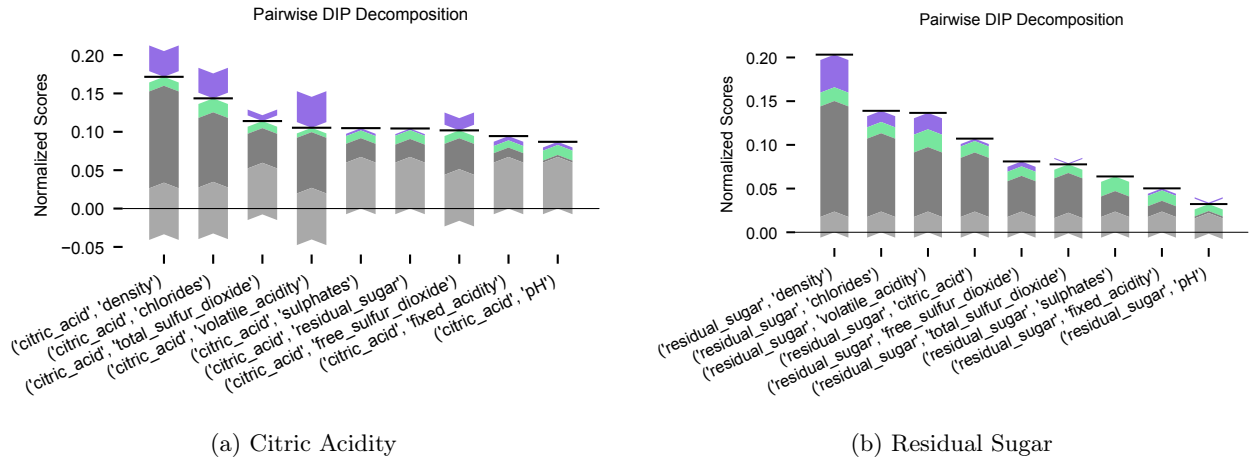


Figure 6: Pairwise DIP decomposition on the Wine Quality Dataset. The two gray bars indicate the standalone contributions of each of the features in the pair, the black horizontal lines indicate the total performance than can be achieved with all features. The interaction surplus is highlighted green, the main effect dependencies purple.

In Section 8, we found that the variable *citric acidity* has a large standalone contribution, but is largely redundant with the remaining features. Furthermore, we found that main effect dependencies between *residual sugar* and the remaining variables have a positive impact on the performance. In this section, we leverage pairwise DIP decompositions to better understand their relationship with specific individual variables.

First, we have a closer look at the role of *citric acidity*. We use DIP to understand which specific variables the feature shares information with (Figure 6a). The decomposition reveals that *citric acidity* has a large negative contribution of main effect dependencies when combined with *volatile acidity*, *density*, and *chlorides*, indicating that the variables have similar roles for the target.

Moreover, we are interested in the role of *residual sugar* for the quality of a wine. We recall that the main effect dependencies between the variable and the remaining variables had a positive impact on the joint performance (and the LOCO score, Figure 3). Using pairwise DIP decompositions (Figure 6b) we find a large positive effect of the main effect dependencies for the pairing with *density*, *chlorides*, and *volatile acidity*. In other words, the DIP decompositions indicate that in these three pairings the variables have opposing relationships with target that are only revealed when analyzed jointly.³ As follows, we focus on analyzing the relationship with *density* using exploratory data analysis.

³We note that in the pairwise DIP, interactions play a more prominent role. This was not the case in the DIP decomposition of the LOCO scores. The reason is that while we previously analyzed the role of residual sugar when added to the remaining variables, we now analyze the role of the variable when added to just one other feature. In this bivariate setting, interactions are necessary to explain what previously could be explained with other variables.

C.1.2 Residual Sugar and Density Have Opposing Effects that Cancel Out

As follows, we have a closer look at the relationship between density, residual sugar, and wine quality. More specifically, we compare a pairwise histogram plot between a feature and target with the plot when additionally conditioning on the respective other feature.

While quality decreases with density (correlation coefficient -0.30 , Figure 7a), no clear trend can be observed when looking at the pairwise relationship between sugar and quality (correlation coefficient -0.03 , Figure 7c). When conditioning on sugar, the negative relationship between density and quality becomes more pronounced (mean correlation over all bins -0.36 , Figure 7b). A positive relationship between sugar and quality becomes visible when conditioning on density (mean correlation over all bins 0.17 , Figure 7d). This suggests that sugar and density have opposing effects on the target but are positively correlated (coefficient 0.55), such that their effects (partially) cancel each other out unless analyzed jointly.

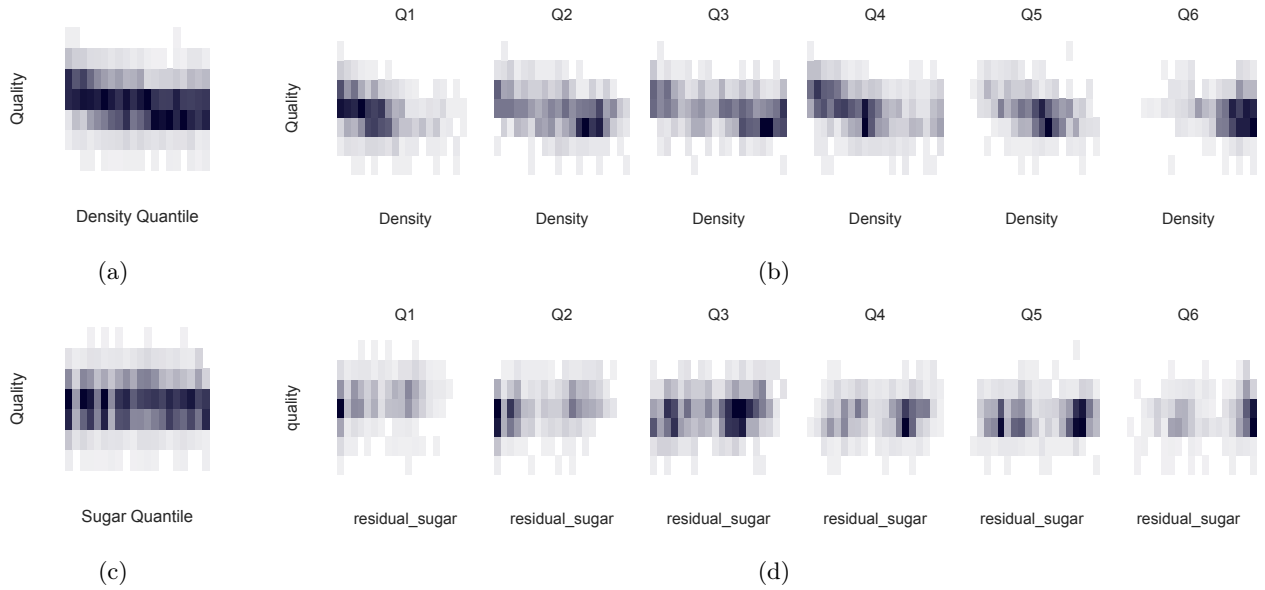


Figure 7: Histogram plots showing the pairwise relationship of density and quality (top row), and sugar and quality (bottom row). To allow a visualization using a heatmap, we first encode the two features as ordinal variables with twenty levels. All levels have the same probability mass and correspond to equal-size quantile ranges. The left plots (Figure 7a and Figure 7c) show the pairwise relationship between each of the features and the target. The right plots (Figure 7b and 7d) show the relationship when conditioning on the respective partner. More specifically, to visualize the conditional density using a histogram plot, we discretize the conditioning variable into six equally sized bins (Q1 to Q6), and create one histogram plot for each bin. In both cases, the variable’s relationship with the target becomes more visible when conditioning on the cooperation partner.

C.2 DIP Decomposition of SAGE on the Wine Quality and California Housing Datasets

The DIP decomposition cannot only be applied to LOCO, but more generally to any explanation technique that is based on comparing the predictive power for different sets of features ($v(S \cup T) - v(S)$). One prominent example are so-called SAGE values.

As follows, we decompose SAGE values using DIP. For their computation we sampled 100 orderings of the features, which each define which feature is added to which coalition of other features. Then, the SAGE value for a feature j is the mean surplus $v(C \cup j) - v(C)$ achieved over the coalitions C implied by the orderings. Each surplus $v(C \cup j) - v(C)$ can be decomposed into a standalone contribution and the cooperative impact. The final SAGE value is the standalone contribution plus the average cooperative impact (details in Appendix A.3).

The results are presented in Figure 8. We apply DIP to the cooperative impact for each coalition, and present the respective average scores for [interaction surplus](#) and [main effect dependencies](#).

First, we have a look at the wine dataset. The feature *citric acidity* again receives a relatively small SAGE

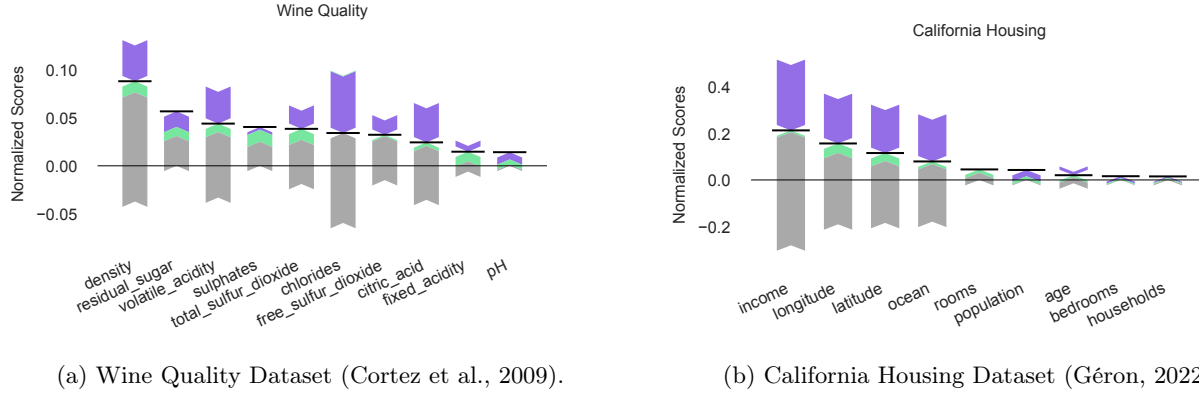


Figure 8: DIP Decompositions of SAGE values (as introduced in Appendix A.3). The grey bars show the features standalone contributions, the purple bars indicate the contributions of dependencies, the green bars the contributions of interactions.

score. The DIP decomposition reveals that the feature is relevant standalone, but receives a small score due to its redundancy with the remaining features. The feature *residual sugar* has a high SAGE score. The DIP decomposition reveals that it has little standalone contribution, but contributes via interactions and positive main effect dependencies.

In the California housing dataset, we again observe that *longitude* and *latitude* are, to a large degree, important due to interactions. The feature *ocean proximity* has a large standalone contribution, but receives a relatively small SAGE score due to its redundancy with the remaining features.

C.3 Additional Datasets

In addition to the two datasets presented in the main paper, we apply the DIP decomposition to LOCO scores on the online news popularity dataset (Fernandes et al. (2015), $d = 62$, $n = 39797$) and the superconductivity dataset (Hamidieh (2018), $d = 81$, $n = 21263$), both taken from the UCI ML repository (Dua and Graff, 2017). We use explainable boosting machines (Nori et al., 2019) as GGAMs and employ a 10-fold cross-validation scheme. Details on estimation, implementation, and computational cost are reported in Appendix B.

Online News Popularity The goal is to predict the number of shares of online articles on a platform called Mashable. The R2 score of the predictor is 0.033. The DIP decompositions of the LOCO scores are presented in Figure 9. According to the LOCO scores, the features *kw_avg_avg* (the average number of shares per keyword averaged over the keywords in the article), *num_videos* (the number of videos), *kw_max_avg* (the maximum number of shares per keyword averaged over all keywords in the article) and *n_unique_tokens* (the rate of unique tokens) are most important, and for example *LDA_03* (the closeness of the topic to LDA 3) is not important. DIP reveals that *LDA_03* indeed has a large standalone contribution, but receives a low LOCO score due to its redundancy with the remaining features. *kw_avg_avg*, and *kw_max_avg* have large standalone contributions and are partly redundant with the remaining features. Furthermore one may think that *n_unique_tokens* is highly predictive of the number of shares; DIP reveals that it has a small standalone contribution but is considered important by LOCO due to the contribution of interactions.

Superconductivity The goal is to predict the critical temperature of superconducting materials. The R2 score of the predictor is 0.924. The results are plotted in Figure 10. We observe that all variables have LOCO scores close to zero, and one may erroneously conclude that all variables are irrelevant for predicting Y . DIP reveals that many features have large standalone relevances but are redundant with the remaining variables due to dependencies between features. Interactions do not contribute to the LOCO scores in this dataset.



Disentangling Interactions and Dependencies in Feature Attribution

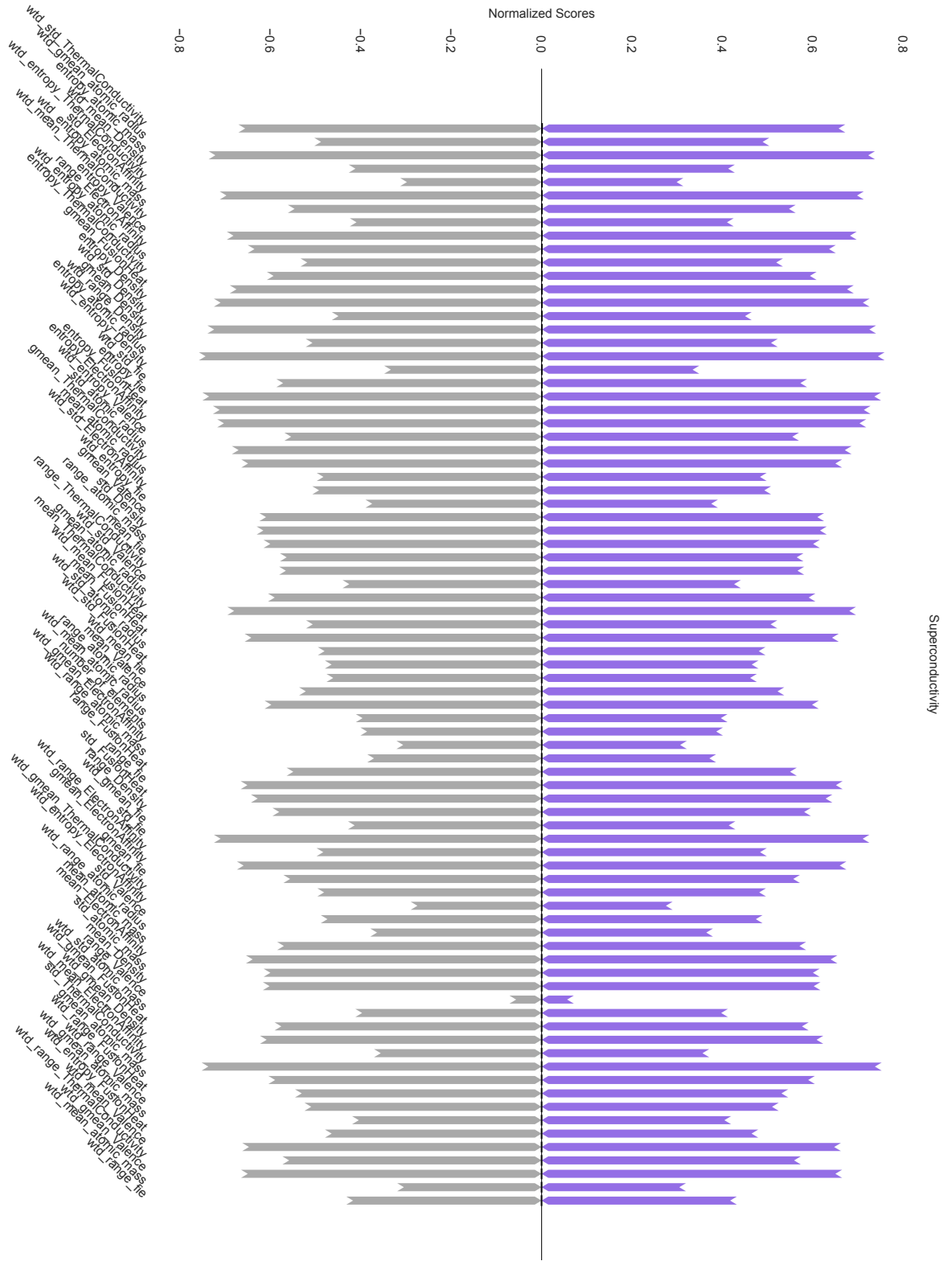


Figure 10: DIP decomposition of the LOCO scores on the superconductivity dataset (Hamidieh, 2018). The grey bars indicate the feature’s standalone contributions, purple bars the contributions of dependencies, and green bars the contributions of interactions. The bars sum up to the LOCO scores, indicated with black horizontal bars.

D Use Case: Using DIP to Assess Whether Features Must Be Interpreted Jointly

Throughout the paper, we argued that features commonly cooperate via dependencies and interactions, meaning they must be interpreted jointly to understand their role for the prediction target. However, many explanation techniques try to explain each feature in isolation. Over the course of this section, we demonstrate how DIP can be used to assess whether such methods reflect the role of the feature in a multivariate context.

Therefore, we first recall a popular interpretation technique that explains the features in isolation. Then we show that the method generally fails, but that DIP identifies cases where it can be applied.

The Conditional Partial Dependence Plot (cPDP) One popular explanation tool that tries to explain features individually is the so-called conditional partial dependence plot (cPDP), also referred to as M-Plot (Apley and Zhu, 2020). The cPDP plots the restricted functions that we defined in Section 3.2. That is, the cPDP for a feature X_j summarizes the role of X_j by integrating out all remaining features using the conditional distribution

$$f_j(x_j) := E(f(x_j, X_{\bar{j}}) \mid X_j = x_j).$$

We recall that for $\mathcal{L}^2(P)$ -optimal predictors f the restricted function is equivalent to the $\mathcal{L}^2(P)$ -optimal univariate predictor $E(Y \mid X_j)$. As such, the cPDP for feature j explains the bivariate relation of the feature with the target and does not take the interplay with the remaining features into account.

Illustrative Examples: cPDPs Are Not Faithful If Features Cooperate In general, explanations such as the cPDP do not reflect the role of the variable in a multivariate model. First of all, if dependencies are present, the effect of a feature on the outcome in a multivariate model, as represented for example in the GGAM component g_j , can change depending on which other variables are included in the model. Second, if there are interactions, the role of the feature may not only change depending on which variables are included in the model, but also depending on the values they take. In both these settings, we must look at the cooperating features jointly to understand the role of the variable for the underlying target. Let us illustrate this with an example.

Example 14. We consider three data generating processes. For each dataset we present a plot of the restricted function f_j (cPDP) (Apley and Zhu, 2020), a plot of the respective GGAM component g_j , and plots that show for each observation x' how the prediction $f(x')$ changes when we vary the value of the feature of interest x_j while keeping the remaining values fixed ($f(x') - f(x_j, x'_j)$, ICE curves (Friedman, 2001)) The plots can be found in Figure 11.

- DGP 1 ($Y \rightarrow X_1 \leftarrow X_2$): $Y \sim N(0, 1)$, $X_2 \sim N(0, 1)$, $\epsilon_1 \sim N(0, 0.2)$, $X_1 := Y + X_2 + \epsilon_1$.
- DGP 2 ($X_1 \rightarrow Y \leftarrow X_2$): $X_1 \sim N(0, 1)$, $X_2 \sim N(0, 1)$, $\epsilon_Y \sim N(0, 0.2)$, $Y := X_1 X_2 + \epsilon_Y$.
- DGP 3 ($X_1 \rightarrow Y \leftarrow X_2$): $X_1 \sim N(0, 1)$, $X_2 \sim N(0, 1)$, $\epsilon_Y \sim N(0, 0.2)$, $Y := X_1 + X_2 + \epsilon_Y$

In DGP 1 dependencies are present such that the bivariate and multivariate relationships between feature and target differ. Specifically, the feature X_2 is pairwise independent of Y but becomes dependent on Y conditional on X_1 . As such, due to the dependencies with the remaining features, the role of the feature for the target depends on which other variables are included in the model. This is reflected in the Figure 11 (a), left, where we see that the restricted model f_2 (cPDP) differs from the GGAM component g_2 : While f_2 (cPDP) is flat, g_2 has a strong negative slope. To understand this negative slope, we have to regard the two features together: In this DGP, the role of X_2 is to denoise the feature X_1 .

In DGP 2, there are no dependencies and f_2 and g_2 coincide. However, since there is an interaction, the f_2 and g_2 differ from the effect of changing the value of x_2 for a specific fixed value of x_1 . This can be seen in Figure 11 (a), center, where we additionally plot the effect of changing the feature X_2 for every observation in the dataset (ICE curves (Friedman, 2001), represented in grey). Thus, to understand the role of the variable for the target, the value of the remaining features need to be taken into account in the interpretation.

Only In DGP 3, where there are no interactions and no dependencies, the cPDP f_2 , the GGAM component g_2 , and observation-wise effects (ICE curves) coincide.

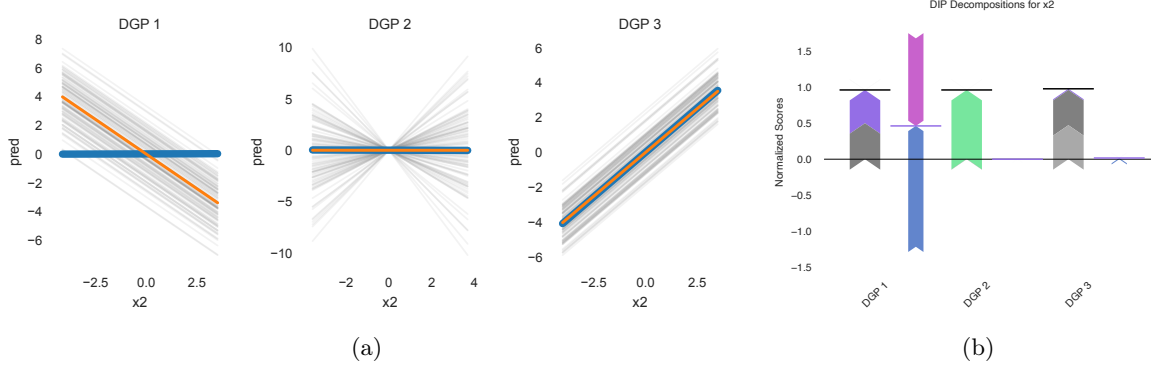


Figure 11: (a) We plot the cPDP in blue, the GGAM component in orange, and the ICE curves in grey. (b) The DIP decompositions for the three DGPs. Grey bars indicate the features' standalone contributions, purple bars the relevance of dependencies (and more specifically of covariance and cross-predictability, and green the contributions of interactions.

DIP Reveals Whether the cPDP Reflects the Role of the Feature As we show in Proposition 15, DIP can distinguish these scenarios: If the cross-predictability score is zero, we know that the univariate predictor and the GGAM component coincide. If the interaction score is zero, we know that the GGAM component and the ICE curves are the same.

Proposition 15. *Let $(X, Y) \sim P$ be a data generating process on $\mathbb{R}^d \times \mathbb{R}$, $j \in D$ the feature of interest, $g^* = g_j^* + g_{\bar{j}}^*$ the $\mathcal{L}^2(P)$ -optimal GGAM in X_j and $X_{\bar{j}}$, and f_j^* the restriction function of f^* , where f^* is the $\mathcal{L}^2(P)$ -optimal predictor of Y given X .*

1. *If the cross-predictability score $CP(j, \bar{j}) = 0$, then f_j^* and g_j^* coincide up to a constant. More precisely, it holds that $f_j^* = g_j^* + E(g_{\bar{j}}^*)$.*
2. *If $Int(j, \bar{j}) = 0$, it holds that $f^*(x'_j, x_{\bar{j}}) - f^*(x_j, x_{\bar{j}})$ is independent of $x_{\bar{j}}$.*

Proof.

1. Since $CP(j, \bar{j}) = \text{Var}(E(g_j^* | X_{\bar{j}})) + \text{Var}(E(g_{\bar{j}}^* | X_j)) = 0$, we know that $E(g_j^* | X_j)$ is constant, which implies $E(g_j^* | X_j) = E(g_j^*)$. We compute

$$f_j^* := E(f^* | X_j) = E(g_j^* + g_{\bar{j}}^* + h^* | X_j) = g_j^* + E(g_{\bar{j}}^* | X_j) + E(h^* | X_j) = g_j^* + E(g_{\bar{j}}^*).$$

2. As $Int(j, \bar{j}) = \text{Var}(h^*) = 0$, we know that $h^* = 0$ because h^* is mean-centered. We compute

$$\begin{aligned} f^*(x'_j, x_{\bar{j}}) - f^*(x_j, x_{\bar{j}}) &= g_j^*(x'_j) + g_{\bar{j}}^*(x_{\bar{j}}) + h^*(x'_j, x_{\bar{j}}) \\ &\quad - g_j^*(x_j) - g_{\bar{j}}^*(x_{\bar{j}}) - h^*(x_j, x_{\bar{j}}) \\ &= g_j^*(x'_j) - g_j^*(x_j). \end{aligned}$$

□

Coming back to our example, we see that the DIP decomposition in Figure 11 (b) distinguishes the three settings. DIP reveals a large cross-predictability score for DGP1 and zero for the remaining DGPs; Indeed, we observe that f_j and g_j only diverge in DGP1. Furthermore, DIP reveals that interactions are important in DGP2 but irrelevant in the remaining DGPs; Indeed, the ICE curves only differ for DGP2.