

On Grouping and Partitioning Approaches in Interpretable Machine Learning

Julia Herbinger

München 2023



On Grouping and Partitioning Approaches in Interpretable Machine Learning

Julia Herbinger

Dissertation

an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

eingereicht von
Julia Herbinger
am 22.09.2023

Erster Berichterstatter: Prof. Dr. Bernd Bischl
Zweiter Berichterstatter: Prof. Dr. Daniel Apley
Dritter Berichterstatter: Prof. Dr. Marvin N. Wright

Tag der Disputation: 05.12.2023

Acknowledgments

I want to express my deepest gratitude to the many wonderful human beings who supported, guided and advised me along this exciting journey. Their help made this thesis possible, so I would like to sincerely thank ...

- ... Prof. Dr. Bernd Bischl and Dr. Giuseppe Casalicchio for their unwavering support, guidance, and mentorship throughout this journey. Their expertise, patience, and dedication have been invaluable in shaping my research and academic growth.*
- ... Prof. Dr. Daniel Apley and Prof. Dr. Marvin N. Wright for agreeing to act as the second and third reviewers for my thesis.*
- ... Prof. Dr. Christian Heumann and PD Dr. Fabian Scheipl for their availability to be part of the examination committee.*
- ... Julia Moosbauer and Quay Au for the excellent and delightful collaboration in our joint projects.*
- ... my fellow researchers and collaborators, both within and outside my institution, for the intellectual discussions, for their collaborative efforts and knowledge-sharing, which enhanced the quality of my work. In particular, I would like to thank Maximilian Muschalik, Fabian Fumagalli, Kristin Blesch, Jan Kapar, Lisa Wimmer, Cornelia Gruber, Theresa Stüber, Philipp Kopper, and Julia Moosbauer. Your presence in my academic and personal life has been a source of strength and motivation. The laughter, shared challenges, and mutual support have made this Ph.D. journey more meaningful.*
- ... my colleagues from Greenpeace and the Munich Poledance community, the camaraderie, dedication, and shared enthusiasm have profoundly impacted my personal and academic growth.*
- ... my dear friends, particularly Katalin, Claudia, Lena, Antonia, Mike, and Christian, for their true friendship and the moments of respite during this challenging journey. Your support helped me maintain a healthy work-life balance.*
- ... my family, particularly my mum Monika, my dad Roland, and my sister and best friend Melanie, your unwavering love, understanding, and belief in my abilities have been my constant motivation.*

Summary

Due to their flexibility in modeling complex relationships such as non-linear and interaction effects, machine learning models are used in various application fields. However, the flexibility also comes with a black box character of these models, leading to a lack of explanation and transparency.

To that end, the research field *interpretable machine learning* to explain the inner workings of machine learning models has grown immensely in recent years. Therefore, model-agnostic interpretation methods that aim to elucidate the influence of features on any model's prediction or performance have been introduced on a global and local level. Global explanations provide insights into the general model behavior. In contrast, local explanations focus on single observations.

While model-agnostic interpretation methods provide more insights into machine learning models, they also bear the risk of being incorrectly applied or interpreted. The first contributing article in Part II of this thesis provides an overview of potential pitfalls, possible solutions, and open issues of existing model-agnostic interpretation methods. The following two parts of this thesis focus on two major limitations of global interpretation methods:

1. human-incomprehensibility of high-dimensional output, and
2. misleading interpretations of global explanations due to aggregation.

Part III of this thesis aims to analyze and propose potential solutions for the limitation *human-incomprehensibility of high-dimensional output*. Most existing global interpretation methods, such as feature importance scores and feature effect visualizations, were introduced on a single-feature level. However, the output of these methods might be overwhelming for high-dimensional settings. This problem can be addressed by grouping features and thus lower dimensionality to simplify the methods' output. Therefore, the second contributing article provides an overview of existing methods and suggests new approaches to quantify feature importance scores and visualize feature effects for feature groups.

Part IV of this thesis is supported by three contributing articles and deals with the limitation of *misleading interpretations of global explanations due to aggregation*. Global feature effect methods may be affected by an aggregation bias due to heterogeneity in local feature effects. This part bridges the gap between local and global feature effect methods by suggesting approaches that provide regional explanations that are more representative of the underlying observations. The heterogeneity in local feature effects is usually either caused by feature interactions or by extrapolation. In the first and second contributing articles of this part, we provide solutions to find interpretable regions based on a recursive partitioning algorithm for cases where the aggregation bias in global feature effect methods is caused by feature interactions. The third contributing article deals with the aggregation bias if it is caused by extrapolating in unseen or sparse regions of the feature space. This article addresses the problem for partial dependence plots when applied to explain hyperparameter effects in the context of hyperparameter optimization. Again, a recursive partitioning algorithm is used to obtain interpretable regions with more confident partial dependence estimates.

Zusammenfassung

Dank ihrer Flexibilität bei der Modellierung komplexer Zusammenhänge wie nichtlinearer und Interaktionseffekte finden maschinelle Lernmodelle in einer Vielzahl von Anwendungsbereichen Verwendung. Diese Flexibilität geht jedoch auch mit einem “Black-Box”-Charakter dieser Modelle einher, was zu einem Mangel an Erklärbarkeit und Transparenz führt. Infolgedessen hat das Forschungsgebiet “Interpretierbares maschinelles Lernen”, das Einblicke in die innere Funktionsweise von maschinellen Lernmodellen ermöglicht, in den letzten Jahren erheblich an Bedeutung gewonnen. Somit wurden modellagnostische Interpretationsmethoden eingeführt, um den Einfluss von Merkmalen auf die Vorhersage oder Performance eines beliebigen Modells sowohl auf globaler als auch auf lokaler Ebene zu erklären. Globale Erklärungen bieten Einblicke in das allgemeine Modellverhalten, während lokale Erklärungen einzelne Beobachtungen erklären. Obwohl modellagnostische Interpretationsmethoden tiefere Einblicke in maschinelle Lernmodelle bieten, bergen sie auch das Risiko, falsch angewendet oder interpretiert zu werden. Der erste beitragende Artikel in Teil II dieser Dissertation bietet einen Überblick über potenzielle Fallstricke, mögliche Lösungen und noch offene Herausforderungen bestehender modellagnostischer Interpretationsmethoden. Die folgenden beiden Teile dieser Arbeit konzentrieren sich auf zwei wesentliche Limitationen globaler Interpretationsmethoden:

1. Die Unverständlichkeit hochdimensionaler Ausgaben
2. Irreführende Interpretationen globaler Erklärungen aufgrund von Aggregation.

Teil III dieser Arbeit zielt darauf ab, die Limitation der *Unverständlichkeit hochdimensionaler Ausgaben* zu analysieren und potenzielle Lösungen vorzuschlagen. Die meisten bestehenden globalen Interpretationsmethoden, wie Merkmalswichtigkeitsbewertungen und Visualisierungen von Merkmalseffekten, wurden auf einzelner Merkmalsebene eingeführt. Die Ausgabe dieser Methoden kann jedoch in hochdimensionalen Merkmalsräumen überwältigend sein. Daher bietet der zweite Beitrag einen Überblick über bestehende Methoden und schlägt neue Ansätze zur Quantifizierung der Merkmalswichtigkeit und Visualisierung von Merkmalswirkungen für Merkmalsgruppen vor.

Teil IV dieser Dissertation wird von drei beitragenden Artikeln unterstützt und behandelt die Limitation *irreführender Interpretationen globaler Erklärungen aufgrund von Aggregation*. Globale Merkmalseffektmethoden können durch Heterogenität in lokalen Merkmalseffekten von einer Aggregationsverzerrung betroffen sein. Dieser Teil der Arbeit schließt die Lücke zwischen lokalen und globalen Merkmalseffektmethoden, indem er Ansätze vorschlägt, die regionale Erklärungen bieten, die repräsentativer für die zugrundeliegenden Beobachtungen in den Regionen sind. Die Heterogenität in lokalen Merkmalseffekten kann verschiedene Ursachen haben. In den ersten beiden Beiträgen dieses Teils bieten wir Lösungen zur Identifizierung interpretierbarer Regionen auf der Grundlage eines rekursiven Partitionierungsalgorithmus an, wenn die Aggregationsverzerrung in globalen Merkmalseffektmethoden durch Interaktionen zwischen Merkmalen verursacht wird. Der dritte beitragende Artikel behandelt das Problem der Aggregationsverzerrung, wenn sie durch Extrapolation in nicht oder kaum gesehenen Regionen des Merkmalsraums verursacht wird. Der Artikel adressiert dieses Problem im Kontext der Erklärung von Hyperparameter-Effekten bei der Hyperparameter-Optimierung mittels Partial Dependence Plots. Erneut wird ein rekursiver Partitionierungsalgorithmus verwendet, um interpretierbare Regionen mit zuverlässigeren regionalen Partial Dependence Schätzungen zu erhalten.

Contents

I. Introduction and Background	1
1. Introduction	3
1.1. Motivation and Scope	3
1.2. Outline	5
2. Background	7
2.1. General Notation and Supervised Machine Learning	7
2.2. Interpretable Machine Learning	8
2.2.1. Overview	8
2.2.2. Model-Agnostic Interpretation Methods	9
2.3. Feature Interactions and Functional ANOVA Decomposition	12
2.3.1. Feature Interactions	12
2.3.2. Functional ANOVA Decomposition	14
2.4. Global Feature Effects	17
2.4.1. Motivation	17
2.4.2. Methodology	17
2.5. Global Feature Importance	21
2.5.1. Motivation	21
2.5.2. Methodology	21
2.6. Limitations of Global Interpretation Methods	23
2.7. Marginal-based versus Conditional-based Approaches	26
II. Pitfalls in Interpretable Machine Learning	29
3. General Pitfalls of Model-Agnostic Interpretation Methods for Machine Learning Models	31
III. Grouping Approaches in Interpretable Machine Learning	63
4. Grouped Feature Importance and Combined Features Effect Plot	65
IV. Partitioning Approaches in Interpretable Machine Learning	117
5. REPID: Regional Effect Plots with implicit Interaction Detection	119
6. Decomposing Global Feature Effects Based on Feature Interactions	145
7. Explaining Hyperparameter Optimization via Partial Dependence Plots	205
V. Conclusion and Open Challenges	219
8. Conclusion	221

9. Open Challenges	223
9.1. Open Challenges of Grouping Approaches	223
9.2. Open Challenges of Partitioning Approaches	223
9.3. General Open Challenges of IML	224
References	225

Part I.

Introduction and Background

1. Introduction

1.1. Motivation and Scope

In recent years, the substantial growth of digital data and the simultaneous improvements in computational power have contributed to notable achievements in artificial intelligence (AI) applications. Billions of people are confronted with AI-based applications on a daily basis when they check their recommendations on social media (Wong, 2023) or for the next series to watch on Netflix (Stoll, 2023) and by using the search engine of Google (Bianchi, 2023). The incredible potential of AI-based applications, particularly of generative AI and large language models, was recognized at the latest when OpenAI released ChatGPT (OpenAI, 2022). The International Data Corporation (IDC) predicted in 2022 that the global value of the AI market will reach 900 billion US Dollars in 2026 (Plachy and Vavra, 2022). According to Bloomberg, the generative AI market alone will increase its value to 1.3 trillion US Dollars within the next ten years, which was around 40 billion in 2022 and thus constitutes a market growth of 3250% (Catsaros, 2023). Given these promising forecasts, the relevance of AI for the future economy and society is undeniable.

With machine learning (ML) algorithms performing extremely well due to their capability to learn complex relationships from data, their application in various fields such as healthcare (Topol, 2019), finance (Heaton et al., 2017), and education (Peters, 2018) is no longer dispensable. However, their ability to flexibly learn complex relationships is based on opaque algorithms. Thus, these algorithms have a black box character, i.e., the inner workings of these algorithms cannot be understood by the user. Consequently, interpretability, which I define according to Miller (2019) by “the degree to which an observer can understand the cause of a decision”, is needed when the decisions made by these black box algorithms affect human life or society. Hence, interpretability is particularly needed if potential biases have been learned by the model and the resulting discriminating actions might have severe consequences for individuals or specific socioeconomic subgroups, such as an unexplainable diagnosis for a life-threatening disease, an unjustified denial of a loan application at a bank or an unjustified risk assessment for a conviction in criminal justice (Carvalho et al., 2019; Gilpin et al., 2018; Watson, 2022). Another important field of application that aspires for interpretability is related to safety-critical tasks such as autonomous driving, where wrong decisions made by an AI system can be perilous for other road users (Gilpin et al., 2018). Since these scenarios are not just hypothetical but have already occurred (Angwin et al., 2016; Obermeyer et al., 2019), the need for interpretability increased. Being able to test, audit, and debug the algorithm before deployment and receiving explanations for potentially wrong decisions can help improve the system and make it more secure (Gilpin et al., 2018). As a result, initial measures have been taken, such as introducing explainability guidelines in the European General Data Protection Regulation (GDPR) (Goodman and Flaxman, 2017). Furthermore, Gartner predicted in 2022 that one of ten major technology trends that will affect the business priorities of organizations in the following three years is “AI Trust, Risk and Security Management”, which “combines methods for explaining AI results, rapidly deploying new models, actively managing AI

security, and controls for privacy and ethics issues” (Groombridge, 2022). Hence, methods that provide insights into these black boxes are required. Therefore, the research area of *interpretable machine learning* (IML), which provides methods that make the predictions of ML models more understandable to humans (Watson, 2022), has received increasing attention and has proliferated in recent years. These interpretation methods can generally be categorized into model-based and post hoc methods (Murdoch et al., 2019). While the former approach pursues to fit directly an interpretable model instead of a black box, the latter approach gains insights into the learned relationships of a black box model by applying these methods to a trained ML model. This thesis focuses on post hoc model-agnostic interpretation methods, which can be applied to any ML model.

Post hoc methods allow us to derive interpretations after training a high-performing ML model to gain insights into the inner workings of the trained ML model. Hence, to obtain interpretability, the model’s performance is not reduced, for example, by using an interpretable or less flexible model class that cannot learn the underlying complex relationships given by the data.¹ While this characteristic holds considerable resonance within the broader research community, post hoc methods also have their pitfalls, which are discussed in detail in the contributing article of Part II of this thesis. The sources of these pitfalls can be categorized into (1) using an unsuitable ML model, (2) the applied IML method is itself limited, and (3) the IML method is not correctly applied (Molnar et al., 2022). While all of these pitfalls need to be taken into consideration when applying post hoc IML methods, the remaining part of the thesis focuses on the second source for model-agnostic interpretation methods. To be more exact, it deals with the limitations of global model-agnostic methods, which try to explain the inner workings of the ML model in general for the entire data set.

Two major limitations of these methods and suggested solutions, which are presented in the contributing articles of Parts III and IV of this thesis, can be summarized as follows:

1. *Human-incomprehensibility of high-dimensional output.* Most global interpretation methods are defined at a single feature level, which makes these methods in high-dimensional settings not only computationally expensive but also incomprehensible for humans when confronted with hundreds or thousands of numbers or visualizations. Part III of this thesis addresses this problem by suggesting several methods to quantify feature importance scores and to create feature effect visualizations for groups of features to lower the dimensionality and thus simplify the resulting output of the interpretation method.

2. *Misleading interpretations of global explanations due to aggregation.* Many global interpretations are estimated by averaging over underlying local interpretations, and thus, their calculation might cause an aggregation bias due to heterogeneity in the underlying local interpretations. This heterogeneity is usually either caused by (1) feature interactions or by (2) extrapolation into sparse or unseen regions of the feature space. In the first two contributing articles of Part IV of the thesis, we suggest solutions for cause 1 by partitioning the feature space such that feature interactions in the resulting regions are minimized, and thus, global interpretation methods (here we focus on feature effect methods) are more representative for the local interpretations in each region. In the third contributing article of Part IV, we address cause 2 of this limitation for partial dependence (PD) plots in the context of explaining hyperparameter effects in automatic ML systems. Here,

¹It should be noted that black box models do not always perform better than interpretable models, but it depends on the complexity of the underlying relationships to be learned by the model (Rudin, 2019). Thus, it is recommended to choose an interpretable model if its performance is similar to that of a black box model (Molnar et al., 2022).

we partition the hyperparameter space such that regions with high uncertainty (sparse regions) are separated from regions with low uncertainty (high-density regions), which results in at least one region that provides a confident and reliable regional PD estimate.

1.2. Outline

The thesis is structured as follows: Section 2 defines and discusses the background knowledge required for the subsequent contributing articles. There, supervised ML is defined, and a general notation is introduced (Section 2.1), followed by a formal definition of IML and a categorization of the methods. A particular focus is on global model-agnostic post hoc interpretation methods (Section 2.2). Then, feature interactions and the functional ANOVA decomposition are defined (Section 2.3). They enhance a better understanding of the following sections, which introduce the most popular global feature effect (Section 2.4) and importance (Section 2.5) methods and explain their limitations (Section 2.6). The final background section covers a general discussion about marginal-based versus conditional-based IML methods (Section 2.7). The background section is followed by three parts, which contain the contributing articles of this thesis. Part II gives an overview of the general pitfalls of model-agnostic IML methods, as well as potential solutions and open challenges to each of these pitfalls. Part III deals with the limitation of the high-dimensional output of interpretation methods by using grouping approaches. Part IV, which is supported by three contributing articles, addresses the aggregation bias of global feature effect methods and provides solutions based on partitioning approaches. The final part of this thesis (Part V) summarizes the main contributions and discusses open challenges regarding the addressed limitations and general limitations of IML methods.

2. Background

2.1. General Notation and Supervised Machine Learning

In ML, users are typically confronted with a p -dimensional feature space $\mathcal{X} = (\mathcal{X}_1 \times \dots \times \mathcal{X}_p)$. In supervised ML problems, we are also given a target space \mathcal{Y} , which is one-dimensional for regression tasks, while its dimensionality depends on the number of classes in a classification task. The respective random variables are then denoted by $X = (X_1, \dots, X_p)$ and Y . We draw n samples i.i.d. of these random variables based on their joint probability distribution $\mathbb{P}_{X,Y}$ to obtain the data set $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$. Generally, we denote $\mathbf{x}^{(i)} = (x_1^{(i)}, \dots, x_p^{(i)})^\top$ and $\mathbf{x}_j = (x_j^{(1)}, \dots, x_j^{(n)})^\top$ to be the feature values of the i -th observation and the j -th feature, respectively.

In supervised settings, a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ maps the feature space to the target space. Supervised ML algorithms are based on a so-called learner or inducer \mathcal{I} which strives to learn the underlying relationship between the features and the target by applying \mathcal{I} on \mathcal{D} leading to the final fitted model $\hat{f} = \mathcal{I}(\mathcal{D})$ (see Figure 2.1). Therefore, the learning algorithm aims to minimize the generalization error measured by the empirical risk $\mathcal{R}_{emp} = \mathbb{E}(L(\hat{f}(X), Y))$, which is defined by the expected loss of a fitted model. To measure the generalization error, the empirical risk needs to be evaluated on an independent test data set¹ \mathcal{D}_{test} that follows the same joint distribution $\mathbb{P}_{X,Y}$ as \mathcal{D} . The estimate of the generalization error of $\hat{f} = \mathcal{I}(\mathcal{D})$ is then calculated by

$$\hat{\mathcal{R}}_{emp}(\hat{f}, \mathcal{D}_{test}) = \frac{1}{|\mathcal{D}_{test}|} \sum_{(\mathbf{x}, y) \in \mathcal{D}_{test}} L(\hat{f}(\mathbf{x}), y). \quad (2.1)$$

In ML we distinguish between white-box and black box models. White-box models are inherently interpretable, i.e., a user understands how a model arrived at a specific prediction. Examples of inherently interpretable models are linear models or (shallow) decision trees. Black box models, on the other hand, are more complex ML algorithms such as (deep) neural networks or gradient boosting algorithms, which often outperform white-box models since they can learn the underlying relationships in the data more flexibly and locally. However, this leads to a more complex model structure that is not inherently interpretable. Therefore, these models are known as black box models.

This thesis only considers supervised ML problems.² Furthermore, the focus in the following sections is on black box ML models. Therefore, the terms supervised ML, ML, and black box are used interchangeably unless explicitly stated otherwise.

¹While the definition in Eq. (2.1) refers to evaluating the generalization error on a holdout test data set, it is more common to use cross-validation to assess the generalization error for ML algorithms (Hastie et al., 2009).

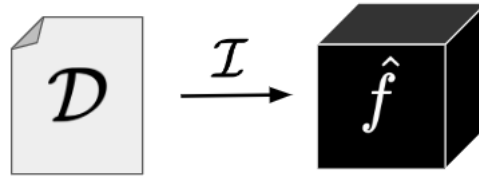


Figure 2.1.: The learner \mathcal{I} is applied on a data set \mathcal{D} to obtain the fitted ML model \hat{f} .

2.2. Interpretable Machine Learning

While users might be satisfied with the predictive performance of the resulting ML model found in the learning process, they are often not able to understand which features the ML model considered in the learning process and how these features influence the final predictions of the ML model. The lack of interpretation might lead to a lack of trust in these models, and depending on the underlying application, this could potentially constitute a decisive factor. The research field IML addresses this concern and provides various solutions to either circumvent the black box character of ML models or to generate insights into the black box. Therefore, I first define IML and categorize the different approaches in the field. Since the contributions in Part II to IV belong to the category of model-agnostic interpretation methods, a more fine-grained classification and description of these methods follow the general overview.

2.2.1. Overview

Murdoch et al. (2019) define IML “as the use of machine-learning models for the extraction of relevant knowledge about domain relationships contained in data”. The way we extract the knowledge, which of it is considered relevant, and how the final output of the interpretation is presented (e.g., by visualizations, formulas, or text) depends on the given data, context, questions, and the respective audience to whom the final results are addressed to. For example, when a model uses “inadmissible” features such as gender or ethnicity, the underlying algorithm might have learned a potential discrimination (Fisher et al., 2019). In medical diagnoses, a doctor will ask different questions about the potentially underlying discriminatory bias than a data engineer in the context of image classification. Hence, different information is required, leading to different interpretation methods to provide this information (Murdoch et al., 2019).

According to Murdoch et al. (2019), IML methods can be grouped into model-based and post hoc interpretation methods as illustrated in Figure 2.2. The main idea of model-based interpretability is to learn a model that is intrinsically interpretable. Thus, we construct a more transparent, “whiteish” model instead of learning a black box model. We can furthermore distinguish between directly fitting inherently interpretable models on \mathcal{D} (such as linear models or decision trees) and fitting an ML model with interpretability constraints. Examples of interpretability constraints include sparsity constraints – fewer features might lead to less complex learned relationships and provide more comprehensible results – or constraints on feature interactions or monotonicity.

However, the simplified definition stated here is more suitable for explaining the fundamentals of the methods covered in this background section.

²Consequently, this thesis does not consider other areas of ML, such as unsupervised ML or reinforcement learning.

Compared to inherently interpretable models, interpretability constraints in ML models might lead to better predictive performance since it possibly allows for modeling the underlying complex relationships in the data more flexibly. However, depending on the chosen model and constraints, interpretability might be limited. Hence, the trade-off between predictive performance and interpretability remains (Du et al., 2019).

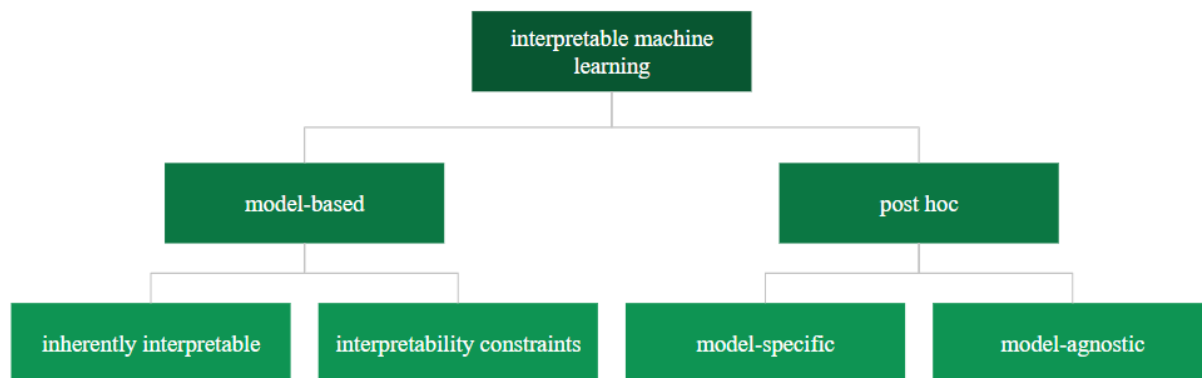


Figure 2.2.: Categorization of IML methods.

Post hoc interpretation methods, on the other hand, do not influence the learning process but are applied after fitting the ML model \hat{f} as illustrated in Figure 2.3. Hence, unlike model-based interpretability, we do not try to learn an intrinsically interpretable model, which may reduce predictive performance. Instead, we learn an arbitrarily complex black box model and apply *post hoc* interpretation methods on the final model to generate insights about the inner workings of the fitted model. We can distinguish between model-specific and model-agnostic *post hoc* interpretation methods (see Figure 2.2). Model-specific methods are developed for a specific ML algorithm. Examples include various methods developed particularly for neural networks (see, e.g., Bach et al., 2015; Selvaraju et al., 2017) or for tree-based methods (see, e.g., Lundberg et al., 2020; Breiman, 2001). While model-specific methods can be optimized for one specific algorithm and use the internal model structure to make computations more efficient, they usually depend on the particular characteristics of this algorithm and cannot be applied to other ML algorithms. Hence, comparing explanations based on a model-specific interpretation method is not possible between different learning algorithms. In contrast, model-agnostic interpretation methods can be applied to any ML algorithm. In the following section, I provide a more detailed overview of model-agnostic interpretation methods. The subsequent sections cover the methodological background of model-agnostic interpretation methods that are most relevant to this thesis and point out the respective limitations that are addressed in the contributing articles of Part III and IV.

2.2.2. Model-Agnostic Interpretation Methods

Model-agnostic interpretation methods are characterized by (1) being applied post hoc on an already fitted black box model (see Figure 2.3) and (2) being independent of the learning algorithm, which means that these methods can be applied to any ML algorithm. These characteristics are often beneficial since model-agnostic interpretation methods can be applied after the model is trained and thus do not require a definition of interpretation goals beforehand. Furthermore, we

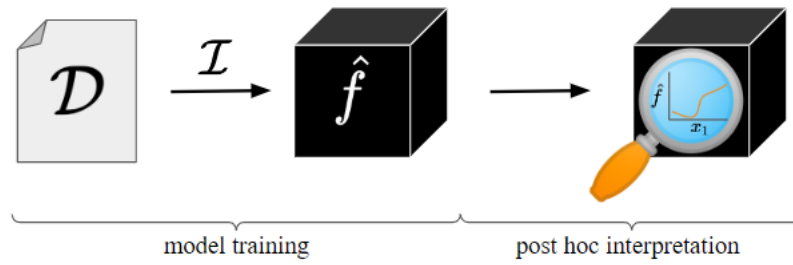


Figure 2.3.: Post hoc interpretability methods are applied to \hat{f} after the model training process.

can compare results between different learning algorithms since the same interpretation method can be applied to different algorithms. Model-agnostic interpretation methods achieve this flexibility as follows: First, data set \mathcal{D} is manipulated (e.g., by perturbing or removing a feature), then based on the manipulated data set, predictions or performance values based on the model \hat{f} are produced³ and compared to the metrics achieved without manipulation (Scholbeck et al., 2020).

Model-agnostic interpretation methods can be categorized depending on different dimensions. I define two dimensions that are most relevant to this thesis and assign the existing methods accordingly. I restrict the type of explanation to feature attributions, which aim to explain the contributions of features regarding the model’s predictions, performance, or variance. Hence, the following categorization does not consider other types of explanations, such as data attributions⁴.

Level of explanation We can distinguish model-agnostic interpretation methods based on which level of explanation they inspect. The two known categories are *local* and *global* interpretation methods (Adadi and Berrada, 2018; Murdoch et al., 2019; Carvalho et al., 2019; Schwalbe and Finzel, 2023).

- *Local* interpretation methods explain single predictions and thus answer questions such as “Which feature was most responsible for the bank’s algorithm to decline the credit application of person XY?”.
- *Global* interpretation methods, on the other hand, aim to explain the general behavior of an ML model and thus generate a general understanding of its inner workings with regard to the given data distribution. Hence, considering the above-stated example, we would answer the question, “Which feature has the highest impact on the model’s decision to decline a credit application based on the given data set?”. Thus, instead of explaining one specific instance of our data set, we try to explain an average instance based on the data.

³Depending on the data manipulation, refitting the model might be necessary, which is discussed in more detail in Section 2.5.

⁴Data attributions analyze the influence of the data instances (observations) on the model’s outcome. For example, Ghorbani and Zou (2019) suggest using Shapley values for data valuation. Thus, the players of the cooperative games are the n data instances instead of the p features.

Type of feature attribution While the taxonomy of local and global interpretation levels is commonly used, there is no consensus on the taxonomy of categorizing feature attribution methods according to the type of explanation they provide.⁵ In this thesis, I categorize model-agnostic feature attribution methods into *feature effect*, *feature importance*, and *feature interaction* methods.

- *Feature effect* methods answer the question “How does a feature of interest influence the predicted outcome of the ML model?”. Hence, feature effects show the direction of the feature’s influence on the model’s prediction. For instance, in the case of the credit scoring example, one could ask if the age of the regarded person has a positive or a negative influence on the predicted credit score of person XY (local explanation).
- *Feature importance* methods, on the other hand, answer the question “How important is a feature?”. The importance, therefore, can relate to the predicted outcome, to the model’s predictive performance or prediction variance. Here, I focus on feature importance methods that measure importance with respect to the model’s predictive performance. Thus, we receive a ranking of features based on how much performance we lose if we lose the information contained in the feature. Considering the credit scoring example, we could ask questions like “Does the model’s expected predictive performance rely on the feature gender?”, or in other words, “How much does the model’s expected predictive performance drop if we remove the information contained in the feature gender?” (global explanation).
- *Feature interaction* methods quantify the share of the contribution of a feature of interest on the model’s prediction that not solely depends on the feature of interest itself but also on other features. For instance, considering the credit scoring example, feature interactions address questions like: “Does the influence of feature age on the predicted credit score differ for male applicants compared to female applicants?” (global explanation). Feature interactions are closely related to feature effects. The relationship is explained in more detail in Sections 2.3, 2.4, and 2.6.

All three feature attribution types can be defined on a local as well as on a global level. Existing model-agnostic interpretation methods can be categorized accordingly, as illustrated in Table 2.1. Popular examples for local feature effect methods are individual conditional expectation (ICE) curves (Goldstein et al., 2015), local interpretable model-agnostic explanations (LIME) (Ribeiro et al., 2016) and Shapley (Štrumbelj and Kononenko, 2014) or Shapley additive explanations (SHAP) values (Lundberg and Lee, 2017). Partial dependence (PD) (Friedman, 2001), accumulated local effects (ALE) (Apley and Zhu, 2020) or SHAP dependence (Lundberg et al., 2020) are well-known representatives for global feature effect methods. While feature importance on a global level can be quantified by permutation feature importance (PFI) (Fisher et al., 2019), leave-one-covariate-out (LOCO) importance (Lei et al., 2018) or Shapley additive global importance (SAGE) (Covert et al., 2020), individual conditional importance (ICI) (Casalicchio et al., 2019) can be used as a local importance measure. Feature interactions on a local level can be visualized by ICE or derivative ICE (d-ICE) curves⁶ and quantified by Shapley-based interaction

⁵For example, Murdoch et al. (2019) distinguishes between feature importance, feature interactions, visualizations, and analyzing trends for global interpretations. In contrast, Adadi and Berrada (2018) categorizes feature attribution methods into visualizations and influence methods. Molnar et al. (2022) distinguishes between feature effect and feature importance methods for local and global methods, which is most similar to the categorization used in this thesis.

⁶These visualizations can be interpreted on a local as well as on a global level.

values (Grabisch and Roubens, 1999; Tsai et al., 2023; Bordt and von Luxburg, 2023; Lundberg et al., 2020). The H-Statistic introduced by Friedman and Popescu (2008) is the most popular approach to quantifying feature interactions on a global level. Further approaches are Greenwell’s interaction index, regional effect plots with implicit interaction detection (REPID), and generalized additive decomposition of global effects (GADGET), which were introduced by Greenwell et al. (2018), and in the contributing articles of Sections 5 and 6, respectively. The list of methods in Table 2.1 is incomplete but contains the most relevant ones for this thesis.

Feature attribution			
	Featue effect	Feature importance	Feature interaction
Level	Local	ICE LIME Shapley/SHAP values	ICI Shapley-based interaction ICE/d-ICE
	Global	PD ALE SHAP dependence	PFI LOCO SAGE H-Statistic Greenwell’s interaction REPID/GADGET

Table 2.1.: Categorization of popular model-agnostic interpretation methods according to the type of feature attribution and the level of explanation. The table is based on the categorization in Molnar et al. (2022) but extended by feature interactions.

The list of methods in Table 2.1 shows that various model-agnostic interpretation methods exist to inspect single predictions and understand the general black box model behavior. However, there also exist several pitfalls one has to be aware of when applying these methods, as demonstrated in the contributing article of Part II of this thesis. Since the remaining contributing articles of this thesis focus on the limitations of global model-agnostic interpretation methods (as described in Section 1.1), I will introduce the relevant methodology of these methods and elaborate on their limitations in the subsequent sections.

2.3. Feature Interactions and Functional ANOVA Decomposition

2.3.1. Feature Interactions

The influence of a feature \mathbf{x}_j on the model’s predictions can be broken down into main and higher-order (interaction) effects of feature \mathbf{x}_j . The main effect of feature \mathbf{x}_j is the influence on the model’s predictions that solely depends on this feature and is independent of the influence of other features \mathbf{x}_{-j} (where $-j = \{1, \dots, p\} \setminus j$ indexes all features except the j -th feature). The higher-order effect of \mathbf{x}_j is the influence of \mathbf{x}_j on the model’s predictions that does not only depend on feature values of \mathbf{x}_j but also on feature values of features in \mathbf{x}_{-j} . Hence, if the feature \mathbf{x}_j does not interact with any other feature in \mathbf{x}_{-j} , the prediction function can be additively decomposed into the main effect of feature \mathbf{x}_j , denoted by $h_j(\mathbf{x}_j)$, and the remaining effect denoted by $h_{-j}(\mathbf{x}_{-j})$: $\hat{f}(\mathbf{x}) = h_j(\mathbf{x}_j) + h_{-j}(\mathbf{x}_{-j})$. The remaining effect is independent of \mathbf{x}_j , and thus only depends on features in \mathbf{x}_{-j} (Friedman and Popescu, 2008). If this additive decomposition is not possible, meaning that $\hat{f}(\mathbf{x}) - h_j(\mathbf{x}_j) - h_{-j}(\mathbf{x}_{-j}) \neq 0$, then interaction effects between feature \mathbf{x}_j and features in \mathbf{x}_{-j} have been learned by the ML model \hat{f} .

Based on this definition, Friedman and Popescu (2008) introduced the H-Statistic to quantify feature interactions on a global level. The H-Statistic between a feature of interest \mathbf{x}_j and all other features \mathbf{x}_{-j} is defined by

$$\mathcal{H}_j^2 = \frac{\sum_{i=1}^n \left(\hat{f}^c(\mathbf{x}^{(i)}) - h_j^c(\mathbf{x}_j^{(i)}) - h_{-j}^c(\mathbf{x}_{-j}^{(i)}) \right)^2}{\sum_{i=1}^n \left(\hat{f}^c(\mathbf{x}^{(i)}) \right)^2}, \quad (2.2)$$

with the superscript c denoting the mean-centered version of the individual or joint effects (e.g., $\hat{f}^c(\mathbf{x}^{(i)}) = \hat{f}(\mathbf{x}^{(i)}) - \frac{1}{n} \sum_{i=1}^n \hat{f}(\mathbf{x}^{(i)})$). Friedman and Popescu (2008) suggest to estimate the feature effect functions h_j and h_{-j} using the PD function, which is introduced in Section 2.4. The H-Statistic value \mathcal{H}_j^2 can be interpreted as the proportion of the model’s prediction variance that is attributable to feature interactions between feature \mathbf{x}_j and all other features in the data (i.e., \mathbf{x}_{-j}). If we calculate the H-Statistic for all features \mathbf{x}_j where $j \in \{1, \dots, p\}$, then we obtain a ranking of all features according to their overall interaction strengths with other features. Since the H-Statistic values in Eq. (2.2) are always scaled by the model’s prediction variance, the interaction values between different features are comparable.

While Eq. (2.2) can be used to detect features that are highly interacting with other features, it does not reveal with which other features the feature \mathbf{x}_j interacts, which orders of interactions have been learned and how strong the effect of each of these higher-order terms are. However, the formula of the H-Statistic in Eq. (2.2) can be adjusted to quantify interactions of different orders. The most common one and easiest to access is the H-Statistic definition for two-way interactions. Following from the definition above, two features \mathbf{x}_j and \mathbf{x}_k do not interact if their joint effect can be additively decomposed into their respective main effects: $h_{jk}(\mathbf{x}_j, \mathbf{x}_k) = h_j(\mathbf{x}_j) + h_k(\mathbf{x}_k)$. Thus, the two-way H-Statistic value can be calculated by

$$\mathcal{H}_{jk}^2 = \frac{\sum_{i=1}^n \left(h_{jk}^c(\mathbf{x}_j^{(i)}, \mathbf{x}_k^{(i)}) - h_j^c(\mathbf{x}_j^{(i)}) - h_k^c(\mathbf{x}_k^{(i)}) \right)^2}{\sum_{i=1}^n \left(h_{jk}^c(\mathbf{x}_j^{(i)}, \mathbf{x}_k^{(i)}) \right)^2}. \quad (2.3)$$

The two-way H-statistic value \mathcal{H}_{jk}^2 can be calculated using the 2-dimensional and the 1-dimensional PD functions (Friedman and Popescu, 2008). The value can be interpreted as the proportion of the variance of the 2-dimensional mean-centered PD of \mathbf{x}_j and \mathbf{x}_k that can be attributed to the interactions between these two features and thus cannot be explained by the main effects of the two features. The scaling factor in the denominator in Eq. (2.3) is based on the joint effect of the two features of interest and thus varies for different features of interest. It follows that the proportions and, therefore, the final H-Statistic values are not comparable between different pairs of features if main effect sizes (proportions of main effects versus interaction effects) differ (Herbinger et al., 2022). Moreover, the H-statistic suffers from potentially wrong rankings due to correlations between the two features of interest (Herbinger et al., 2022). Another global interaction measure to quantify two-way interactions based on PDs has been suggested by Greenwell et al. (2018). While the ranking of feature interactions of their approach is not affected by feature correlations between the two features of interest, it is also sensitive with regard to the main effect sizes. In our contributions of Section 5, we show theoretically and empirically how these two pitfalls affect the ranking of feature interactions for the H-Statistic and the approach by Greenwell et al. (2018) and suggest an alternative interaction ranking measure for two-way interactions based on PD and ICE functions, which does not suffer from the same disadvantages. In the contributing article

of Section 6, we generalize this approach to other feature effect methods such as ALE or SHAP dependence.

While most introduced global feature interaction methods focus on quantifying two-way interactions, Hooker (2004) introduced an algorithm that also detects feature interactions of higher order than two. The resulting feature relationships are visualized in a network graph. However, the method does not quantify or rank feature interactions. Quantifying all higher-order terms requires decomposing the prediction function into all main and higher-order effects of all features. These decompositions are desirable from an interpretation perspective but are usually challenging to estimate. In the following section, I introduce – from an IML perspective – the most popular functional decomposition, the functional ANOVA decomposition, and discuss the benefits and underlying challenges of this approach.

2.3.2. Functional ANOVA Decomposition

The functional ANOVA decomposition has, amongst others, been studied and extended for decomposing prediction functions by Stone (1994); Hooker (2004, 2007); Rahman (2014). If the prediction function of an ML model is square-integrable, we can use the functional ANOVA decomposition to decompose it into the main and higher-order effects of involved features as follows:

$$\hat{f}(\mathbf{x}) = g_0 + \sum_{j=1}^p g_j(\mathbf{x}_j) + \sum_{j \neq k} g_{jk}(\mathbf{x}_j, \mathbf{x}_k) + \dots + g_{12\dots p}(\mathbf{x}) = \sum_{k=1}^p \sum_{\substack{W \subseteq \{1, \dots, p\}, \\ |W|=k}} g_W(\mathbf{x}_W), \quad (2.4)$$

with g_0 representing an additive constant (comparable to an intercept in a linear model), $g_j(\mathbf{x}_j)$ representing the main effect of each feature indexed by $j \in \{1, \dots, p\}$ and $g_{jk}(\mathbf{x}_j, \mathbf{x}_k)$ being the pure two-way interaction effects between all pair of features. All effects that cannot be explained by main or pure two-way interaction effects are then assigned to further higher-order effects with a potential p-way remaining effect that cannot be explained by any lower-order effect. Thus, the functional ANOVA decomposition always exactly decomposes the prediction function.⁷

Being able to uniquely decompose the prediction function into all components, meaning into all main and higher-order effects of all features, is desirable from an interpretability perspective since this decomposition provides many insights into how features individually and jointly influence the model’s predictions. For example, if we assume that a bank fitted an ML model to predict the credit score of their customers based on the age (\mathbf{x}_{age}), profession (\mathbf{x}_{prof}), savings (\mathbf{x}_{sav}) and desired loan amount (\mathbf{x}_{amt}) of the applicants, and if we can decompose the predictions of the ML model into the main and higher-order effects of each of the involved features as in Eq. (2.4) by

$$\begin{aligned} \hat{f}(\mathbf{x}) = & g_0 + g_{age}(\mathbf{x}_{age}) + \dots + g_{age,prof}(\mathbf{x}_{age}, \mathbf{x}_{prof}) + \dots + g_{age,prof,sav}(\mathbf{x}_{age}, \mathbf{x}_{prof}, \mathbf{x}_{sav}) \\ & + \dots + g_{age,prof,sav,amt}(\mathbf{x}_{age}, \mathbf{x}_{prof}, \mathbf{x}_{sav}, \mathbf{x}_{amt}), \end{aligned}$$

then we can answer questions like:

⁷A more formal definition of how these components are determined is provided at the end of this section and depends on the restrictiveness of underlying assumptions.

1. How does each feature individually influence the ML model’s predicted outcome? For instance, we might be interested in the main effect of age ($g_{age}(\mathbf{x}_{age})$), i.e., we would like to know how the age of the applicants individually influences the predicted credit score.
2. Which influence on the model’s predictions can only be explained when two features are considered together and thus cannot be explained by their individual (main) effects? For example, we would like to know how the effect of age on the predicted credit score varies depending on the feature profession. If the effect of age changes for different profession categories, then feature interactions between age and profession are present, and it follows $g_{age,prof}(\mathbf{x}_{age}, \mathbf{x}_{prof}) \neq 0$.
3. Similarly to point 2, we can ask questions regarding the effects of a higher order than two. Thus, there might be effects that cannot be explained by only considering the combination of age and profession, but also the amount in the savings account of the applicants changes the influence on the predicted credit score for different combinations of age and profession. In this case the three-way interaction effect $g_{age,prof,sav}(\mathbf{x}_{age}, \mathbf{x}_{prof}, \mathbf{x}_{sav})$ is non-zero.

Overall, the decomposition generates insights into the complexity of learned relationships of the underlying ML model. Suppose a high proportion of the learned effects is due to interaction effects (especially of high order). In that case, the learned relationships are rather complex and local. At the same time, prediction functions that can be decomposed into components of low order (e.g., only main and some two-way interaction effects) are less complex and easier to explain on a global level. The latter case might imply that a simpler and more interpretable model might lead to a similar predictive performance to the chosen more complex black box model.

The following paragraphs define the underlying assumptions and resulting properties of a unique functional ANOVA decomposition and the consequential challenges of estimating such a decomposition. In general, we distinguish between the standard and the generalized functional ANOVA decomposition, of which the former is based on stronger assumptions, which makes it unsuitable in the presence of (strong) feature correlations (Hooker, 2007).

Standard functional ANOVA decomposition The standard functional ANOVA decomposition (Hooker, 2004) assumes that the probability density function is defined by a product-type probability measure $w(\mathbf{x}) = \prod_{j=1}^p w_j(\mathbf{x}_j)$ with $w_j : \mathbb{R} \rightarrow \mathbb{R}_0^+$ denoting the marginal probability density function of feature \mathbf{x}_j . This assumption implies that random variables in X and, thus, the features are independent of each other. Following from that assumption and given that the *vanishing condition*⁸ is fulfilled, we can uniquely and optimally decompose the prediction function \hat{f} into each single component function $g_W(\mathbf{x}_W)$ of Eq. (2.4) (Li and Rabitz, 2012; Rahman, 2014). The *vanishing condition* is defined by

$$\int g_W(\mathbf{x}_W) w_j(\mathbf{x}_j) d\mathbf{x}_j = 0 \quad \forall j \in W \neq \emptyset, \tag{2.5}$$

where $\int_{\mathbb{R}} w_j(\mathbf{x}_j) d\mathbf{x}_j = 1$ and $w_j(\mathbf{x}_j) \geq 0$ holds. Thus, the component functions $g_W(\mathbf{x}_W)$ “integrate to zero with respect to the marginal density of each random variable” in W (Rahman, 2014).

The *vanishing condition* results in two properties: The *zero means* property $\mathbb{E}[g_W(\mathbf{x}_W)] = 0$ and the *orthogonality* property $\mathbb{E}[g_W(\mathbf{x}_W) g_V(\mathbf{x}_V)] = 0$ with $\emptyset \neq W \subseteq \{1, \dots, p\}$, $\emptyset \neq V \subseteq \{1, \dots, p\}$

⁸The *vanishing condition* is also known as strong annihilating condition (Rahman, 2014).

and $W \neq V$. Hence, each component function's expected value (mean) is zero, and two distinct component functions are orthogonal to each other.

These properties allow us to determine the component functions of Eq. (2.4) sequentially:

$$g_W(\mathbf{x}_W) = \int_{\mathbf{x}_{-W}} \left(\hat{f}(\mathbf{x})w(\mathbf{x}) - \sum_{V \subset W} g_V(\mathbf{x}_V) \right) d\mathbf{x}_{-W}. \quad (2.6)$$

Hence, each component function $g_W(\mathbf{x}_W)$ represents the pure W -th order main or interaction effect and is calculated by subtracting all effects of lower order than W from the joint effect of all features in W . This approach is straightforward if the probability density $w(\mathbf{x})$ is defined by a product-type probability measure, which allows us to integrate over marginal rather than joint distributions. However, as mentioned before, this implies that features are independent of each other, which is a strong and often unrealistic assumption in real-world settings.

Generalized functional ANOVA decomposition If the random variables in X are not independently distributed, the probability density $w(\mathbf{x})$ cannot be written as a product of marginal densities, and thus the vanishing condition in Eq. (2.5) does not hold. To facilitate a unique decomposition of the prediction function $\hat{f}(\mathbf{x})$ in the presence of correlated features, Hooker (2007) relaxed the vanishing condition to

$$\int g_W(\mathbf{x}_W)w(\mathbf{x}) d\mathbf{x}_j d\mathbf{x}_{-W} = 0 \text{ for } j \in W \neq \emptyset, \quad (2.7)$$

meaning that the integral over each coordinate direction of the subset W is zero. With $w(\mathbf{x})$ representing a general probability density with its support being grid-closed, the component functions of the generalized functional ANOVA decomposition can, according to Hooker (2007), be uniquely determined by optimizing

$$g_W(\mathbf{x}_W) = \arg \min_{\{h_W \in \mathbb{L}^2(\mathbb{R}^W), W \subseteq \{1, \dots, p\}\}} \int \left(\sum_{W \subseteq \{1, \dots, p\}} h_W(\mathbf{x}_W) - \hat{f}(\mathbf{x}) \right)^2 w(\mathbf{x}) d\mathbf{x}. \quad (2.8)$$

The functional ANOVA decomposition is one approach that guarantees a unique decomposition of the prediction function by preferring lower-order terms over higher-order terms. This idea follows the reluctance principle (Sun et al., 2022) and thus tends to decompose the prediction function into a simple structure that enhances interpretability.

Note that all non-constant component functions have zero means for both the standard and the generalized functional ANOVA decomposition. However, compared to the standard version, the relaxed vanishing condition of the generalized functional ANOVA decomposition only allows for hierarchical orthogonality, meaning only component functions are orthogonal to each other if one is a subset of the other. It follows that the unique decomposition cannot be achieved sequentially anymore, but instead, the more complex and computationally expensive optimization problem in Eq. (2.8) needs to be solved. Recent research addresses this challenge by either including the hierarchical orthogonality condition as a constraint in the modeling process (Sun et al., 2022) or by suggesting more efficient model-specific solutions such as for tree-based models (Lengerich et al., 2020). While these are promising first approaches to estimate the generalized functional ANOVA decomposition efficiently, one should keep in mind that the decomposition itself relies on how well we can approximate the true data distribution (Lengerich et al., 2020).

2.4. Global Feature Effects

2.4.1. Motivation

Feature effects are widely used model-agnostic interpretation methods that address the question of how features influence the model's predictions. While local feature effect methods focus on explaining the features' influence on a single prediction, global feature effects focus on effects that explain the average influence of the (individual) features on the model predictions with respect to the given data distribution. For the linear model, the global feature effect of a feature of interest corresponds to its estimated coefficient multiplied by its feature values. Since the effect is linear, we can interpret the result as follows: If the value of feature \mathbf{x}_j increases by 1, the predicted outcome changes by the size and sign of the estimated coefficient⁹ while keeping all other features constant. Hence, the influence of a feature on the predictions in a linear model can – due to its linearity – be summarized by one number.¹⁰ The influence of features on an ML model's predictions, on the other hand, is often not linear, which makes summarizing the feature effect in one number more challenging. Hence, global feature effect methods in IML are usually based on visualizations.

2.4.2. Methodology

Here, I describe the most popular global feature effect methods, namely the PD, ALE, and SHAP dependence plots. Since the final result is presented in a plot, we usually limit the number of features that are visualized to one or two. For the sake of simplicity and relevance for this thesis, the following definitions and examples are based on one feature of interest \mathbf{x}_j .¹¹ In that case, the plot to visualize the feature effect for the feature of interest shows the feature values of \mathbf{x}_j on the x-axis and the respective predicted feature effects on the y-axis. The global feature effect is then visualized by a curve showing the predicted average feature effect for feature \mathbf{x}_j .

Partial dependence The PD plot (Friedman, 2001) is amongst the most popular global feature effect methods. One reason for its popularity is the intuitive definition and estimation approach, which makes it very accessible to users. The PD function for feature \mathbf{x}_j is defined by

$$f_j^{PD}(\mathbf{x}_j) = E_{X_{-j}}[\hat{f}(\mathbf{x}_j, X_{-j})] = \int \hat{f}(\mathbf{x}_j, \mathbf{x}_{-j}) d\mathbb{P}(\mathbf{x}_{-j}). \quad (2.9)$$

Thus, the PD is the expected marginal effect of feature \mathbf{x}_j , which can be estimated by integrating over the joint distribution $\mathbb{P}(\mathbf{x}_{-j})$ of all other features \mathbf{x}_{-j} . However, we usually do not have access to the joint distribution, and thus, $\mathbb{P}(\mathbf{x}_{-j})$ is approximated by using Monte Carlo integration to estimate the PD function:

$$\hat{f}_j^{PD}(\mathbf{x}_j) = \frac{1}{n} \sum_{i=1}^n \hat{f}(\mathbf{x}_j, \mathbf{x}_{-j}^{(i)}). \quad (2.10)$$

⁹In the presence of multiplicative feature interactions, the predicted outcome changes by the sum of the respective coefficients.

¹⁰To compare features on different scales, we calculate the feature effect by multiplying the coefficient with the respective feature values.

¹¹However, the definitions can easily be extended to more than one feature of interest (Herbinger et al., 2023).

The final PD plot is then created by visualizing $\{(\mathbf{x}_j^{(k)}, \hat{f}_j^{PD}(\mathbf{x}_j^{(k)}))\}_{k=1}^m$ based on m grid points¹². The resulting PD curve is an average over the n local ICE curves $\hat{f}(\mathbf{x}_j, \mathbf{x}_{-j}^{(i)})$ (Goldstein et al., 2015). An ICE curve describes how a feature of interest influences the prediction of a single instance. For example, the upper left plot in Figure 2.5 shows the ICE curve for the feature age of a female passenger of the ocean liner Titanic. The predicted survival probability of the ML model for this woman is 96.7%, which is marked by the dotted line at her age of 19. The orange ICE curve shows how the predicted survival probability of this woman would change if she had a different age, but all her other characteristics, such as her passenger class or the fare price she paid, are fixed. For instance, the predicted survival probability decreases for higher age values than her actual age, leaving all other characteristics unchanged. The right plot of Figure 2.5 shows the ICE curves of all 891 passengers and the PD curve (blue), which is the average over the 891 ICE curves calculated at each grid point. Thus, the PD curve here represents the average marginal effect of the feature age on the predicted survival probability, i.e., it addresses how changing an average passenger’s age affects their expected predicted survival probability. It is also observable that the ICE curves in this plot are very heterogeneous. Thus, the feature age influences the predicted outcome differently for different individuals. This heterogeneity can be explained by feature interactions between the feature of interest (feature age) and other features that are used for modeling – for example, the gender or passenger class of a person (Goldstein et al., 2015; Herbringer et al., 2022). Thus, it is recommended to visualize ICE curves (local effects) and the PD curve (global effect) together in one plot as shown in Figure 2.5 as this provides more insights into the learned effects of a model than only considering the aggregated version (PD plot).

Both the calculation and the interpretation of PD and ICE plots are very intuitive and provide comprehensive insights into how a feature of interest influences the model’s predictions. One reason for this simplicity lies in the assumption that PD functions are calculated by integrating over marginal distributions, which assumes feature independence. Similarly to the standard functional ANOVA decomposition, this assumption allows us to decompose the prediction function sequentially based on the mean-centered PD functions for all possible feature subsets up to a constant (Friedman, 2001). However, using marginal distributions for integration causes extrapolation in unseen regions in the presence of (strong) feature correlations when estimating ICE and PD curves. This extrapolation problem is illustrated in Figure 2.4. Here, features \mathbf{x}_1 and \mathbf{x}_3 are highly correlated with each other, resulting in the regions of small \mathbf{x}_1 combined with large \mathbf{x}_3 values and large \mathbf{x}_1 combined with small \mathbf{x}_3 values that the ML model did not see during the learning process. Thus, the fitted neural network may exhibit an oscillating extrapolation effect when estimating ICE curves in these unseen regions. Since the PD curve averages over all ICE curves, including the ones in the sparsely sampled regions, the PD estimate might not reflect the underlying data distribution very well, as shown in this figure. One possibility to overcome this problem is to use the conditional instead of the marginal distribution in Eq. (2.9), which is also called Marginal (M) plot (Friedman, 2001; Apley and Zhu, 2020). While M plots have the advantage that they do not extrapolate in unseen regions, the interpretation of the feature effect changes. By using the conditional distribution, the M plot reflects not only the feature effect of the feature of interest but also partially the effect of the features correlated with the feature of interest. Hence, the M plot does not allow visualizing the individual marginal effect of the feature

¹²Usually not all feature values of \mathbf{x}_j are used for calculating and visualizing the PD plot but only a smaller number of m grid points, which can be defined as a random sample of feature values, quantile values or an equidistant grid of the feature range of \mathbf{x}_j (Molnar et al., 2022).

of interest and provides no information on how this combined effect can be decomposed into the different correlated features.

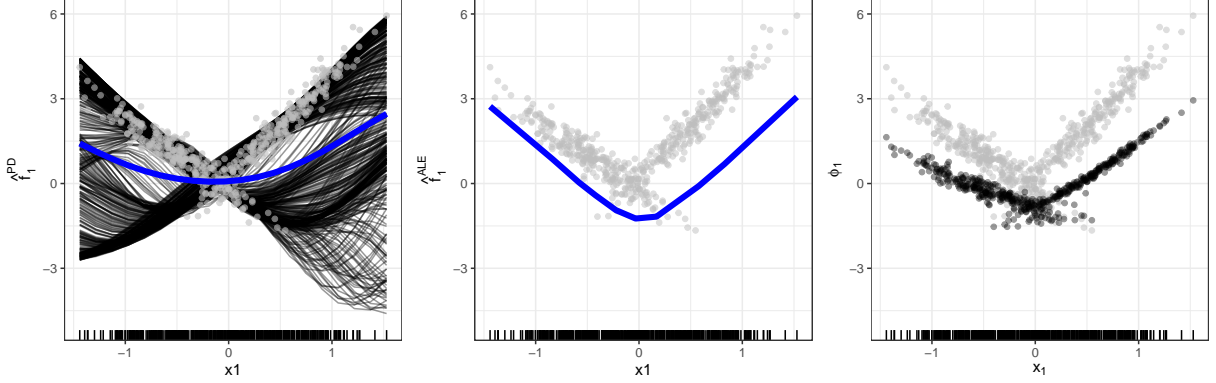


Figure 2.4.: The figure shows the PD and ICE plot (left), ALE plot (middle) and SHAP dependence plot (right) for feature \mathbf{x}_1 of the following simulation example, which is taken from Herbringer et al. (2023): Let $X_2, X_3 \sim \mathcal{U}(-1, 1)$ be independently distributed and $X_1 = X_3 + \delta$ with $\delta \sim \mathcal{N}(0, 0.0625)$. The data generating process is defined by $Y = 3X_1 \mathbb{1}_{X_3 > 0} - 3X_1 \mathbb{1}_{X_3 \leq 0} + X_3 + \epsilon$, with $\epsilon \sim \mathcal{N}(0, 0.09)$. We draw 500 observations and fit a tuned feed-forward neural network¹³ on the data set. It holds for all plots (if available): Blue curves represent the global average feature effect, black curves or points visualize the local feature effects, and grey points are the actual data points. The effect values of ALE and SHAP show an additive shift compared to the actual data points since these effect methods are centered by their means.

Accumulated local effects ALE (Apley and Zhu, 2020) is another global feature effect method, which uses the conditional instead of the marginal distribution for integration and thus does not suffer from the issues extrapolation may induce. ALE functions are calculated such that the effect curve only represents the feature effect of the feature of interest. Hence, compared to M plots, the ALE curve does not reflect the effects of features correlated with the feature of interest. The ALE function $f_j^{ALE}(x)$ for feature \mathbf{x}_j at feature value $x \sim \mathbb{P}(\mathbf{x}_j)$ is given by

$$f_j^{ALE}(x) = \int_{z_0}^x \mathbb{E} \left[\frac{\partial \hat{f}(X)}{\partial X_j} \Big| X_j = z_j \right] dz_j = \underbrace{\int_{z_0}^x \int \underbrace{\frac{\partial \hat{f}(z_j, \mathbf{x}_{-j})}{\partial z_j}}_{(1)} d\mathbb{P}(\mathbf{x}_{-j} | z_j)}_{(2)} dz_j, \quad (2.11)$$

where $z_0 = \min(\mathbf{x}_j)$. ALE is estimated according to the following three steps related to the numbering of Eq. (2.11):

1. Calculate the local derivatives for each observation with respect to its feature value $\mathbf{x}_j = z_j$. A common approach to calculate the local derivatives is to create quantile-based intervals for

¹³More information about the hyperparameter tuning and specifications can be found in Herbringer et al. (2023).

the feature range of \mathbf{x}_j and determine for each observation the prediction difference between the upper and lower boundary of the respective interval that contains z_j .

2. Integrate the local derivatives over the conditional distribution $\mathbb{P}(\mathbf{x}_{-j}|z_j)$. This integral is equivalent to a conditional expectation and is estimated by calculating the mean values of the local derivatives within each interval.
3. The global ALE effect at $\mathbf{x}_j = x$ is then calculated by accumulating the conditional expected values of step (2) up to the interval that contains x .

By calculating the derivatives in step (1), the additive feature effects that do not contain the feature of interest – meaning all main and interaction effects that only consider features in \mathbf{x}_{-j} – are removed. Thus, ALE does not consider the effects of features correlated with \mathbf{x}_j . Note that since the calculation of the final curve in step (3) is based on derivatives, the additive shift cannot be meaningfully interpreted as it is with PD plots. Hence, the mean-centered ALE curve is typically calculated and visualized (Apley and Zhu, 2020).

Similar to PD plots, ALE plots visualize solely the feature effect for the feature of interest. However, compared to PD plots, they do not extrapolate into sparse or unseen regions as illustrated in the plot in the middle of Figure 2.4. Furthermore, Apley and Zhu (2020) showed that the prediction function can be uniquely decomposed by ALE functions up to a constant and that this decomposition satisfies an orthogonality-like property, which is similar to the hierarchical orthogonality property of the generalized functional ANOVA decomposition. One disadvantage of ALE is that the final visualization only shows the global effect curve. Since the global curve is determined by first integrating and then accumulating over the local derivatives, the local effects (derivatives) cannot be meaningfully visualized within the ALE plot. Hence, the final plot does not provide any information about the heterogeneity of underlying local effects.

SHAP dependence Another method that visualizes the influence of a feature on the model’s predictions is the SHAP dependence plot (Lundberg et al., 2020). The SHAP dependence plot is based on SHAP values (Lundberg and Lee, 2017), which are the additive feature attribution definition of Shapley values.¹⁴ Shapley values originate from game theory (Shapley, 1953) to calculate the fair payout for each player in a cooperative game. The general concept was transferred to ML where it is used to fairly distribute the predicted outcome of a single observation to the involved features (Štrumbelj and Kononenko, 2014). Thus, Shapley values are a local feature effect method. They rely on a proper axiomatic foundation, which guarantees that features that do not contribute to the prediction receive a Shapley value of zero (dummy axiom), that the Shapley values of all features sum up to the prediction (efficiency axiom), that features that contribute equally to the prediction receive the same Shapley values (symmetry axiom) and that Shapley values are additive for arbitrarily weighted ensemble of models (additivity axiom). The Shapley value is the unique feature attribution method that fulfills all these axioms and thus allows a fair distribution of the predicted outcome to all features. The Shapley value definition for the feature of interest \mathbf{x}_j is based on its contribution to all possible subsets of the remaining features. Each

¹⁴Based on the additive feature attribution definition, Lundberg and Lee (2017) proposed a more efficient estimation technique of Shapley values called Kernel SHAP. However, theoretically and if calculated exactly, Shapley and SHAP values are the same.

of these feature subsets forms a coalition denoted by $S \subseteq \{1, \dots, p\} \setminus j$. The Shapley value of feature \mathbf{x}_j at feature value x is then defined by

$$\phi_j(x) = \sum_{S \subseteq \{1, \dots, p\} \setminus j} \frac{|S|!(p - |S| - 1)!}{p!} (v(S \cup j) - v(S)), \quad (2.12)$$

with the value function $v(S) = \mathbb{E}_{X_{-S}}[\hat{f}(\mathbf{x}_S, X_{-S})] - \mathbb{E}_X[\hat{f}(X)]$.

To create the SHAP dependence plot, the Shapley (SHAP) values for the feature of interest \mathbf{x}_j need to be calculated for all observations of the data set. The plot then shows the feature values of \mathbf{x}_j on the x-axis and the respective Shapley (SHAP) values of each observation on the y-axis (see the right plot in Figure 2.4). Hence, the SHAP dependence plot is a pendant to the ICE plot and does not provide an average marginal feature effect curve like the PD or ALE curve. However, it visualizes the overall trend and the heterogeneity of local effects and thus indicates if feature interactions between \mathbf{x}_j and other features are present.

While PD plots are based on the marginal feature distributions and ALE plots are based on conditional feature distributions, Shapley values can be calculated based on either of the two. If Shapley values are based on marginal distributions, then the value function of Eq. (2.12) can be defined based on PD functions and thus might also be affected by extrapolation when features are correlated.

Recent research also focused on relating Shapley values and SHAP dependence plot to functional decompositions and thus allow to separate main from higher-order effects (Hiabu et al., 2023; Herren and Hahn, 2022; Bordt and von Luxburg, 2023).

2.5. Global Feature Importance

2.5.1. Motivation

Unlike feature effect methods, feature importance methods are not concerned about how the feature influences the model predictions. Instead, these methods quantify the strength of the feature's influence on the model's predictions, prediction variance, or predictive performance. The importance score for a feature of interest summarizes the strength of the feature's influence in a single number. This quantity can then be used to rank the features according to their importance. In ML, we are typically interested in how important a feature is regarding the model performance (i.e., the generalization error) rather than the model predictions themselves since only considering the predictions does not tell us how well the ML model \hat{f} fits the underlying data. Thus, we would like to understand which features play a crucial role in obtaining a high model performance.

2.5.2. Methodology

Many performance-based global feature importance methods typically rely on either feature permutations or model refits. Permutation-based approaches assess the importance of features by permuting them while keeping the fitted model fixed. In contrast, refitting-based approaches conduct changes in the feature space (e.g., by removing a feature), requiring a refit. I will now

define and discuss popular permutation-based and refitting-based approaches, namely, PFI and LOCO.

Permutation feature importance The PFI was introduced by Breiman (2001) for random forests and generalized to a model-agnostic version by Fisher et al. (2019). The PFI of feature \mathbf{x}_j is defined by the difference between the expected loss (empirical risk) after permuting the j -th feature and the expected loss of the originally fitted model \hat{f} :

$$\text{PFI}_j(\hat{f}) = \mathbb{E}(L(\hat{f}(X_{[j]}), Y)) - \mathbb{E}(L(\hat{f}(X), Y)), \quad (2.13)$$

with $X_{[j]} = (X_1, \dots, X_{j-1}, \tilde{X}_j, X_{j+1}, \dots, X_p)$ representing the random variable vector of features with \tilde{X}_j being a random variable that is independent of X_j but that follows the same marginal distribution as X_j .

The PFI of a feature \mathbf{x}_j is calculated as follows: First, the feature values of \mathbf{x}_j for the test data set are randomly permuted. Second, the model \hat{f} that was fitted on the training data set is applied to the test data set, but feature values of \mathbf{x}_j are exchanged by the randomly permuted version of \mathbf{x}_j . Third, we subtract the empirical risk based on the initial predictions of the test data (without permutation) from the empirical risk calculated based on the predictions of the test data set that incorporates the permuted feature of interest to determine the risk difference for this permutation. The PFI is then calculated by repeating the three steps for several random permutations of \mathbf{x}_j and averaging over the risk differences obtained in step three.¹⁵

The intuition behind permuting the feature values is to break the association between the target variable and the feature of interest, in this example, \mathbf{x}_j . It follows that if the permutation of the feature values does not change the model’s performance, the feature is not considered important. By randomly permuting the feature values of \mathbf{x}_j over all observations in the data set without considering the underlying correlation structure with other features, PFI uses marginal sampling and thus applies a similar logic to PD. Hence, when features are correlated, PFI is also affected by extrapolating in sparse or unseen regions. Conditional variants of PFI can be estimated by using a conditional instead of a marginal sampling strategy for the feature of interest (Freiesleben et al., 2023). Common approaches are model-X knockoffs (Watson and Wright, 2021) or subgroups (Molnar et al., 2023). However, one must be careful when comparing the importance scores of the two strategies since PFI based on marginal sampling, as defined above, answers the question of how important the feature itself (irrespective of all other features) is for the model’s performance. In contrast, PFI based on conditional sampling addresses the question of how much a feature contributes in addition to all other features used by the model (hence conditioned on the remaining features). The different approaches are discussed on a more general level in Section 2.7.

Leave-one-covariate-out importance While permutation-based methods have the advantage that only new predictions based on an already trained model need to be generated, refitting-based methods require refitting the algorithm, which is usually computationally more expensive. The most popular refitting-based feature importance method is LOCO (Lei et al., 2018), which refits the learner \mathcal{I} based on a reduced data set where the feature of interest \mathbf{x}_j has been removed:

¹⁵Note that for an exact computation, all possible random permutation need to be computed. However, it is commonly approximated in practice by choosing a smaller number of random permutations and calculating the PFI by Monte Carlo integration (Casalicchio et al., 2019; Au et al., 2022).

$\tilde{\mathcal{D}} := \{(\mathbf{x}_{-j}^{(i)}, y^{(i)})\}_{i=1}^n$. Based on the original model fit on the entire data set $\mathcal{I}(\mathcal{D}) = \hat{f}_{\mathcal{D}}$ and the model fitted on the reduced data set $\mathcal{I}(\tilde{\mathcal{D}}) = \hat{f}_{\tilde{\mathcal{D}}}$, the LOCO importance score $LOCO_j(\mathcal{I})$ for feature \mathbf{x}_j is defined by

$$LOCO_j(\mathcal{I}) = \mathbb{E}(L(\hat{f}_{\tilde{\mathcal{D}}}(X_{-j}), Y)) - \mathbb{E}(L(\hat{f}_{\mathcal{D}}(X), Y)). \quad (2.14)$$

Hence, LOCO measures the difference between the expected loss when leaving out the feature of interest \mathbf{x}_j and refitting the algorithm and the expected loss of the original model that was fitted to the full set of features. Thus, LOCO calculates the feature importance based on the question: Can we remove the feature \mathbf{x}_j without losing performance if we refit the model based on the remaining features? Hence, we ask if the information contained in the j -th feature that is relevant to the model’s performance can be learned by the remaining features in \mathbf{x}_{-j} when the learner is refitted on the reduced data set. If the performance of the refitted model $\hat{f}_{\tilde{\mathcal{D}}}$ does not drop in expectation compared to the originally fitted model $\hat{f}_{\mathcal{D}}$, the j -th feature is considered irrelevant. Note that with this definition, a feature is considered not only irrelevant (or barely relevant) for the model if it is not used by the model at all, but also if other, potentially highly correlated features are able to reach a similar model performance without the help of the feature of interest. However, if the feature \mathbf{x}_j provides additional valuable information for making predictions on the target, the performance of the refitted model $\hat{f}_{\tilde{\mathcal{D}}}$ is reduced in expectation compared to the performance of originally fitted model $\hat{f}_{\mathcal{D}}$ and thus feature \mathbf{x}_j is considered important in terms of LOCO.¹⁶

To sum up, PFI quantifies the influence of a single feature of interest on the model performance by breaking the association to the target variable and all other features in the data set. Thus, PFI represents the influence of the feature of interest, disregarding the influence of all other features for a fitted model \hat{f} . In contrast, LOCO aims to quantify the additional importance provided by feature \mathbf{x}_j to the model fit on the full data set compared to the model refit on the reduced data set, which does not take into account the information provided by \mathbf{x}_j . Hence, the two feature importance methods provide different information, which needs to be considered when interpreting the results. In the contributing article in Section 4, we discuss the differences in more detail, extend both approaches to feature groups, and show how to leverage the grouped versions to gain more insights into which combination of features are most important in terms of model performance by suggesting a new importance-based sequential procedure. Based on the definitions of the permutation-based and refitting-based importance methods for feature groups, we derive a grouped version of SAGE, which can be calculated by either of the two approaches.

2.6. Limitations of Global Interpretation Methods

After introducing the most popular model-agnostic global interpretation methods, I will now discuss two of their major limitations that are in the focus of this thesis.

¹⁶Note that refitting-based approaches are usually calculated by using a resampling technique, and therefore several refits and evaluations on different data sets are required (Au et al., 2022).

Human-incomprehensibility of high-dimensional output The global feature effects and importance methods introduced in Section 2.4 and 2.5 are usually defined for a single feature of interest. Hence, we can create, for instance, a PD plot for each feature in the data set or calculate an importance score for each feature and provide the user with a table of their importance rankings. Single-feature interpretations usually work well for low-dimensional data sets, typically used for illustration purposes in research articles. However, in real-world applications, we are often confronted with hundreds or even thousands of features that are used for modeling. For example, genetic or sensor applications frequently incorporate high-dimensional data sets that contain complex relationships like feature interactions of high order and highly correlated features such as genes in a given pathway (He and Yu, 2010; Gregorutti et al., 2015) or spectral bands of satellite images in sensor data (Chakraborty and Pal, 2008). In addition to the high computational cost, which is a challenging problem for many interpretation methods (see pitfall 9.2 in the contributing article of Part II), grasping the interrelationships and the overall picture of the resulting high-dimensional output is usually not feasible for humans. Furthermore, feature effect methods like PD plots are visual tools. Thus, they are usually limited to two dimensions, making it difficult to understand if the model has learned interactions of higher order than two and how they influence the model’s predictions.

In the contributing article of Part III of this thesis, we address this limitation and propose solutions for both global feature importance and global feature effect methods. The suggested interpretation methods are defined for groups of features. Therefore, the resulting output is of lower dimensionality, which may increase comprehensibility.

Misleading interpretations of global explanations due to aggregation Global interpretation methods are usually defined as an aggregation over local interpretations (e.g., the PD curve of feature \mathbf{x}_j is an average over the ICE curves of feature \mathbf{x}_j). This aggregation has the advantage that the information is simplified and thus more understandable for the user in the sense that it is easier to interpret one number or curve than n numbers or curves. However, by aggregating quantities across observations, the granularity of information is lost. For most methods, the information loss is particularly high when features interact and when features are correlated, which may lead to what is known as aggregation bias (Mehrabi et al., 2021; Herbinger et al., 2022).

In the following, I will present two concrete examples of how this aggregation bias limits the meaningfulness of the PD plot as an IML method.

Aggregation bias due to feature interactions. For illustration purposes, I use the Titanic data set (Dawson, 1995). This data set contains 11 characteristics of 891 passengers of the ocean liner Titanic. The target variable is the binary label if they survived the disaster, and the chosen ML model is a random forest with 500 trees (Herbinger et al., 2022). Let us assume that we are interested in how the age of the passengers influences their predicted survival probability. Let us furthermore assume that we are interested in two specific passengers of the ocean liner:

1. a female passenger, aged 19, with a first-passenger class ticket and
2. a male passenger, aged 22, with a third-passenger class ticket.

We use ICE curves to visualize the effect of the feature age for the two passengers. The two respective ICE curves are shown in Figure 2.5, where the dotted line marks the predicted survival probability of each passenger. The upper orange ICE curve of the female passenger (1) shows

an overall high predicted survival probability across all age values. However, compared to the actual age of 19, it decreases slightly for older women and young girls. The lower yellow ICE curve of the male passenger (2), on the other hand, shows a different behavior. The predicted survival probability for this passenger (across all age values) is far lower than for passenger 1. Moreover, the influence of the feature age on the predictions is different for the two passengers. For the male passenger, the predicted survival probability stays relatively constant for higher age values. In contrast, it increases strongly for passengers of younger age while keeping the other characteristics constant. Hence, in this model, age does have a different influence on the predicted survival probability for passenger 1 compared to passenger 2. The right plot in Figure 2.5 visualizes the ICE curves of all 891 passengers. The curves are very heterogeneous, indicating that age does have a different influence on the predicted survival probability for different passengers depending on their other characteristics. This heterogeneous behavior of ICE curves indicates that age interacts with other features of the data set, which leads to heterogeneous local feature effects. However, the plot does not tell us which feature interactions have been learned by the model, and due to the high number of very heterogeneous curves, the plot itself becomes incomprehensible. The PD curve (blue) shows the aggregation over all 891 ICE curves. However, as clearly visible, the PD curve is not representative of many of the underlying ICE curves, and thus, basing the interpretations only on the PD curve might lead to misguided conclusions for many individuals.

One solution to that problem might be to estimate the functional ANOVA decomposition and visualize only the main effect of the feature of interest by a PD plot. In this case, interaction effects are separated into additive functions, and the aggregated effect curve for the main effect *does* represent all underlying local effect curves. However, as mentioned in Section 2.3, estimating the functional ANOVA decomposition remains a challenging and computationally expensive task. And even if we could compute the functional ANOVA decomposition, visualizing and interpreting effects of a higher order than two remains an open issue.

Founded on the theoretical concept of functional ANOVA decomposition, we address the described limitation for global feature effect methods and suggest solutions based on recursive partitioning in the first and second contributing articles of Part IV of this thesis. The proposed algorithms partition the feature space such that feature interactions are minimized. Thus, the regional feature effects are more representative of the underlying local effects in the final regions.

Aggregation bias due to extrapolation. In Figure 2.4, I already illustrated that the heterogeneity of ICE curves can also be caused by an oscillating behavior of the algorithm in extrapolating regions. In particular, algorithms that exhibit local instabilities, like neural networks, are affected by this problem. Since the PD function aggregates over all ICE curves (with equal weights) and does not distinguish between dense and sparse regions of the feature space, potentially extreme predictions from extrapolating regions might strongly influence the shape of the PD curve. Thus, the final average marginal effect curve might inappropriately represent the underlying observations.

A concrete application affected by this phenomenon is the interpretation of hyperparameter effects in automated ML, where efficient optimizers, like Bayesian optimization, are applied to find good configurations for the hyperparameters of the ML algorithm to be tuned. These optimizers try to find a good balance between exploring the hyperparameter space and exploiting in regions, which seem promising to converge to a good configuration as fast as possible. Thus, while promising regions for a good configuration usually show a high number of sampled configurations, other less promising regions show only a small number of sampled configurations or none at all. Thus, the underlying model will be uncertain when predicting in these sparse or unseen regions of

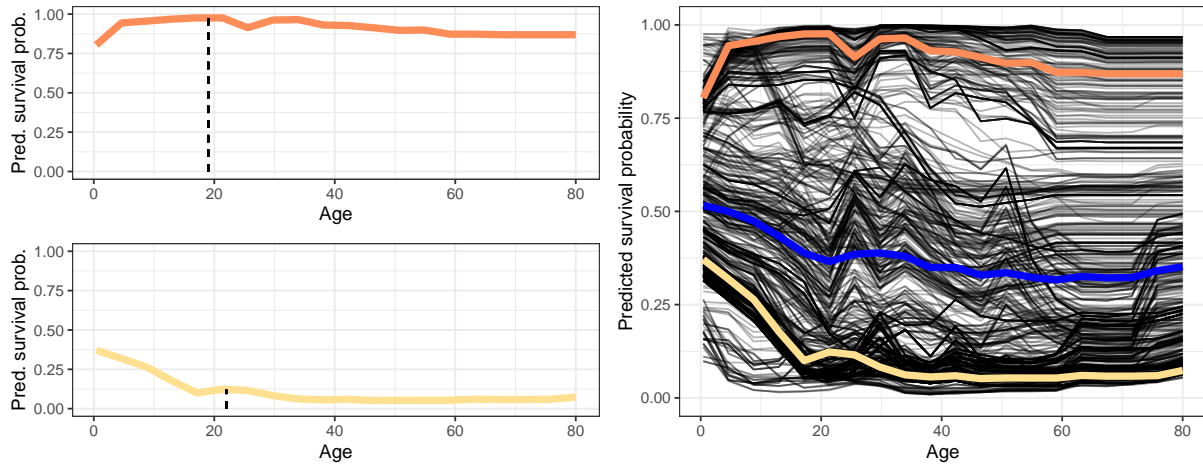


Figure 2.5.: The left plots show the ICE curves for the feature age of two passengers (upper: female, lower: male) from the Titanic data set. The dotted lines mark the prediction of the random forest for their actual age. The right plot visualizes these two ICE curves and all other ICE curves (black) of the remaining passengers, as well as the PD curve (blue) for the feature age.

the hyperparameter space, leading to uncertain and unreliable PD estimates in these regions (Moosbauer et al., 2021).

In the third contributing article of Part IV of this thesis, we analyze the described aggregation bias of PD plots for hyperparameter effects in hyperparameter optimization and suggest a potential solution based on recursive partitioning, which provides more confident and reliable PD estimates for hyperparameter effects in relevant regions of the hyperparameter space.

The aggregation bias due to extrapolation is not only problematic for PD plots but for every global interpretation method based on marginal distributions or sampling strategies such as PFI. Another solution to this limitation that focuses on interpreting the influence of features rather than hyperparameters by PD plots and PFI is suggested by Molnar et al. (2023).

2.7. Marginal-based versus Conditional-based Approaches

In the discussion of various global model-agnostic interpretation methods in the previous sections, the question of whether we should use approaches based on marginal or conditional distribution or sampling strategies arose several times. Recent research criticizes marginal-based approaches because they extrapolate in sparse or unseen regions of the feature space in the presence of dependent features, which might lead to explanations that are based on predictions of unrealistic data points (Aas et al., 2021; Frye et al., 2020). Therefore, some favor the conditional variants of the methods, such as conditional feature importance instead of PFI (Watson and Wright, 2021; Blesch et al., 2023; Molnar et al., 2023), M plots instead of PD plots, or Shapley values based on conditional instead of marginal sampling approaches (Aas et al., 2021). However, the conditional variants have their downsides, too. Some works criticize method-specific disadvantages. For example, features not used by the ML model may receive a non-zero Shapley value if conditional

instead of marginal sampling is used (Janzing et al., 2020; Chen et al., 2020). The main disadvantage is that the conditional distribution needs to be approximated, which becomes especially difficult and computationally expensive when training data is sparse or of high dimensionality (Janzing et al., 2020; Sundararajan and Najmi, 2020). Ultimately, the approximation of the conditional distribution influences the final interpretation and might even lead to counter-intuitive explanations (Sundararajan and Najmi, 2020).

Arguments exist for both perspectives, so the question remains: Which one to choose? I follow the argumentation of Watson (2022), Freiesleben et al. (2022) and Chen et al. (2020) that there is not one approach that fits it all, but that the choice depends on different factors, in particular which question one would like to answer by using IML. Therefore, I distinguish between two general goals:

1. If we are interested in understanding the inner workings of the fitted ML model, marginal approaches may be most suitable.
2. If we are interested in interpretations that reflect the underlying data structures, conditional methods may be most suitable.

The first goal is particularly important in model auditing to debug a model and to understand the influence of features that the model has learned. The second goal is especially interesting when we want to draw inferences from the model (Freiesleben et al., 2022).

Besides these two general goals, users need to be aware that depending on the interpretation method, marginal-based methods are differently interpreted than conditional-based methods. For example, while PFI quantifies the absolute importance of each feature irrespectively of all other features, conditional versions of PFI quantify the additional importance of a feature of interest, considering all other feature values are known. Also, the interpretation of PD and M plots differ. PD plots visualize solely the influence of the feature of interest. In contrast, M plots visualize the influence of the feature of interest and, in parts, the influence of the features correlated with it.

To summarize, the choice between marginal-based and conditional-based approaches depends on the context, the desired interpretation, and whether we are interested in answering questions about the model or the data. However, this choice is only to be made if features are correlated since the two approaches are identical in the case of independent features.

Part II.

**Pitfalls in Interpretable Machine
Learning**

3. General Pitfalls of Model-Agnostic Interpretation Methods for Machine Learning Models

This article reviews the general pitfalls of existing model-agnostic interpretation methods. We summarize suggested solutions and remaining open challenges for each of the pitfalls. These pitfalls can be categorized according to their source into (1) an inappropriate ML model is used, (2) the IML method itself is limited, and (3) the IML method is misapplied. The remaining contributing articles of this thesis propose solutions to the pitfalls categorized in the second source.



Contributing article: Molnar, C., König, G., Herbinger, J., Freiesleben, T., Dandl, S., Scholbeck, C. A., Casalicchio, G., Grosse-Wentrup, M. and Bischl, B. (2022). General pitfalls of model-agnostic interpretation methods for machine learning models. In *A. Holzinger, R. Goebel, R. Fong, T. Moon, K.-R. Müller, and W. Samek (Eds.), xxAI - Beyond Explainable AI, Volume 13200 of Lecture Notes in Artificial Intelligence*, pp. 39–68, Cham: Springer. https://doi.org/10.1007/978-3-031-04083-2_4.

Author contributions: Julia Herbinger contributed to this paper as a co-author with the following significant contributions:

The project idea was developed by Christoph Molnar, Gunnar König, Julia Herbinger, Timo Freiesleben, Susanne Dandl, Christian Scholbeck, and Giuseppe Casalicchio with equal contributions. Christoph Molnar initiated, led, and coordinated the project. The manuscript was drafted jointly by Christoph Molnar, Gunnar König, Julia Herbinger, Timo Freiesleben, Susanne Dandl, Christian Scholbeck, and Giuseppe Casalicchio. Julia Herbinger wrote the chapters “Misleading Interpretations Due to Feature Interactions” and “Human-Intelligibility of High-Dimensional IML Output” with support from Giuseppe Casalicchio. All authors contributed to revisions of the paper and suggested several notable modifications.



General Pitfalls of Model-Agnostic Interpretation Methods for Machine Learning Models

Christoph Molnar^{1,7}, Gunnar König^{1,4}, Julia Herbinger¹,
Timo Freiesleben^{2,3}, Susanne Dandl¹, Christian A. Scholbeck¹,
Giuseppe Casalicchio¹, Moritz Grosse-Wentrup^{4,5,6}, and Bernd Bischl¹

¹ Department of Statistics, LMU Munich, Munich, Germany
christoph.molnar.ai@gmail.com

² Munich Center for Mathematical Philosophy, LMU Munich, Munich, Germany

³ Graduate School of Systemic Neurosciences, LMU Munich, Munich, Germany

⁴ Research Group Neuroinformatics, Faculty for Computer Science,
University of Vienna, Vienna, Austria

⁵ Research Platform Data Science @ Uni Vienna, Vienna, Austria

⁶ Vienna Cognitive Science Hub, Vienna, Austria

⁷ Leibniz Institute for Prevention Research and Epidemiology - BIPS GmbH,
Bremen, Germany

Abstract. An increasing number of model-agnostic interpretation techniques for machine learning (ML) models such as partial dependence plots (PDP), permutation feature importance (PFI) and Shapley values provide insightful model interpretations, but can lead to wrong conclusions if applied incorrectly. We highlight many general pitfalls of ML model interpretation, such as using interpretation techniques in the wrong context, interpreting models that do not generalize well, ignoring feature dependencies, interactions, uncertainty estimates and issues in high-dimensional settings, or making unjustified causal interpretations, and illustrate them with examples. We focus on pitfalls for global methods that describe the average model behavior, but many pitfalls also apply to local methods that explain individual predictions. Our paper addresses ML practitioners by raising awareness of pitfalls and identifying solutions for correct model interpretation, but also addresses ML researchers by discussing open issues for further research.

Keywords: Interpretable machine learning · Explainable AI

This work is funded by the Bavarian State Ministry of Science and the Arts (coordinated by the Bavarian Research Institute for Digital Transformation (bid)), by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A, by the German Research Foundation (DFG), Emmy Noether Grant 437611051, and by the Graduate School of Systemic Neurosciences (GSN) Munich. The authors of this work take full responsibilities for its content.

© The Author(s) 2022

A. Holzinger et al. (Eds.): xxAI 2020, LNAI 13200, pp. 39–68, 2022.

https://doi.org/10.1007/978-3-031-04083-2_4

1 Introduction

In recent years, both industry and academia have increasingly shifted away from parametric models, such as generalized linear models, and towards non-parametric and non-linear machine learning (ML) models such as random forests, gradient boosting, or neural networks. The major driving force behind this development has been a considerable outperformance of ML over traditional models on many prediction tasks [32]. In part, this is because most ML models handle interactions and non-linear effects automatically. While classical statistical models – such as generalized additive models (GAMs) – also support the inclusion of interactions and non-linear effects, they come with the increased cost of having to (manually) specify and evaluate these modeling options. The benefits of many ML models are partly offset by their lack of interpretability, which is of major importance in many applications. For certain model classes (e.g. linear models), feature effects or importance scores can be directly inferred from the learned parameters and the model structure. In contrast, it is more difficult to extract such information from complex non-linear ML models that, for instance, do not have intelligible parameters and are hence often considered black boxes. However, model-agnostic interpretation methods allow us to harness the predictive power of ML models while gaining insights into the black-box model. These interpretation methods are already applied in many different fields. Applications of interpretable machine learning (IML) include understanding pre-emption decision-making [124] with partial dependence plots [36], inferring behavior from smartphone usage [105, 106] with the help of permutation feature importance [107] and accumulated local effect plots [3], or understanding the relation between critical illness and health records [70] using Shapley additive explanations (SHAP) [78]. Given the widespread application of interpretable machine learning, it is crucial to highlight potential pitfalls, that, in the worst case, can produce incorrect conclusions.

This paper focuses on pitfalls for model-agnostic IML methods, i.e. methods that can be applied to any predictive model. Model-specific methods, in contrast, are tied to a certain model class (e.g. saliency maps [57] for gradient-based models, such as neural networks), and are mainly considered out-of-scope for this work. We focus on pitfalls for global interpretation methods, which describe the expected behavior of the entire model with respect to the whole data distribution. However, many of the pitfalls also apply to local explanation methods, which explain individual predictions or classifications. Global methods include the partial dependence plot (PDP) [36], partial importance (PI) [19], accumulated local effects (ALE) [3], or the permutation feature importance (PFI) [12, 19, 33]. Local methods include the individual conditional expectation (ICE) curves [38], individual conditional importance (ICI) [19], local interpretable model-agnostic explanations (LIME) [94], Shapley values [108] and SHapley Additive exPlanations (SHAP) [77, 78] or counterfactual explanations [26, 115]. Furthermore, we distinguish between feature effect and feature importance methods. A feature effect indicates the direction and magnitude of a change in predicted outcome due to changes in feature values. Effect methods include

		Local	Global
Feature	Effects	ICE LIME Counterfactuals Shapley Values SHAP	PDP ALE
	Importance	ICI	PI PFI SAGE

Fig. 1. Selection of popular model-agnostic interpretation techniques, classified as local or global, and as effect or importance methods.

Shapley values, SHAP, LIME, ICE, PDP, or ALE. Feature importance methods quantify the contribution of a feature to the model performance (e.g. via a loss function) or to the variance of the prediction function. Importance methods include the PFI, ICI, PI, or SAGE. See Fig. 1 for a visual summary.

The interpretation of ML models can have subtle pitfalls. Since many of the interpretation methods work by similar principles of manipulating data and “probing” the model [100], they also share many pitfalls. The sources of these pitfalls can be broadly divided into three categories: (1) application of an unsuitable ML model which does not reflect the underlying data generating process very well, (2) inherent limitations of the applied IML method, and (3) wrong application of an IML method. Typical pitfalls for (1) are bad model generalization or the unnecessary use of complex ML models. Applying an IML method in a wrong way (3) often results from the users’ lack of knowledge of the inherent limitations of the chosen IML method (2). For example, if feature dependencies and interactions are present, potential extrapolations might lead to misleading interpretations for perturbation-based IML methods (inherent limitation). In such cases, methods like PFI might be a wrong choice to quantify feature importance.

Table 1. Categorization of the pitfalls by source.

Sources of pitfall	Sections
Unsuitable ML model	3, 4
Limitation of IML method	5.1, 6.1, 6.2, 9.1, 9.2
Wrong application of IML method	2, 5.2, 5.3, 7, 8, 9.3, 10

Contributions: We uncover and review general pitfalls of model-agnostic interpretation techniques. The categorization of these pitfalls into different sources is provided in Table 1. Each section describes and illustrates a pitfall, reviews possible solutions for practitioners to circumvent the pitfall, and discusses open issues that require further research. The pitfalls are accompanied by illustrative

examples for which the code can be found in this repository: https://github.com/compstat-lmu/code_pitfalls_uml.git. In addition to reproducing our examples, we invite readers to use this code as a starting point for their own experiments and explorations.

Related Work: Rudin et al. [96] present principles for interpretability and discuss challenges for model interpretation with a focus on inherently interpretable models. Das et al. [27] survey methods for explainable AI and discuss challenges with a focus on saliency maps for neural networks. A general warning about using and explaining ML models for high stakes decisions has been brought forward by Rudin [95], in which the author argues against model-agnostic techniques in favor of inherently interpretable models. Krishnan [64] criticizes the general conceptual foundation of interpretability, but does not dispute the usefulness of available methods. Likewise, Lipton [73] criticizes interpretable ML for its lack of causal conclusions, trust, and insights, but the author does not discuss any pitfalls in detail. Specific pitfalls due to dependent features are discussed by Hooker [54] for PDPs and functional ANOVA as well as by Hooker and Mentch [55] for feature importance computations. Hall [47] discusses recommendations for the application of particular interpretation methods but does not address general pitfalls.

2 Assuming One-Fits-All Interpretability

Pitfall: Assuming that a single IML method fits in all interpretation contexts can lead to dangerous misinterpretation. IML methods condense the complexity of ML models into human-intelligible descriptions that only provide insight into specific aspects of the model and data. The vast number of interpretation methods make it difficult for practitioners to choose an interpretation method that can answer their question. Due to the wide range of goals that are pursued under the umbrella term “interpretability”, the methods differ in which aspects of the model and data they describe.

For example, there are several ways to quantify or rank the features according to their relevance. The relevance measured by PFI can be very different from the relevance measured by the SHAP importance. If a practitioner aims to gain insight into the relevance of a feature regarding the model’s generalization error, a loss-based method (on unseen test data) such as PFI should be used. If we aim to expose which features the model relies on for its prediction or classification – irrespective of whether they aid the model’s generalization performance – PFI on test data is misleading. In such scenarios, one should quantify the relevance of a feature regarding the model’s prediction (and not the model’s generalization error) using methods like the SHAP importance [76].

We illustrate the difference in Fig. 2. We simulated a data-generating process where the target is completely independent of all features. Hence, the features are just noise and should not contribute to the model’s generalization error. Consequently, the features are not considered relevant by PFI on test data.

However, the model mechanistically relies on a number of spuriously correlated features. This reliance is exposed by marginal global SHAP importance.

As the example demonstrates, it would be misleading to view the PFI computed on test data or global SHAP as one-fits-all feature importance techniques. Like any IML method, they can only provide insight into certain aspects of model and data.

Many pitfalls in this paper arise from situations where an IML method that was designed for one purpose is applied in an unsuitable context. For example, extrapolation (Sect. 5.1) can be problematic when we aim to study how the model behaves under realistic data but simultaneously can be the correct choice if we want to study the sensitivity to a feature outside the data distribution.

For some IML techniques – especially local methods – even the same method can provide very different explanations, depending on the choice of hyperparameters: For counterfactuals, explanation goals are encoded in their optimization metrics [26, 34] such as sparsity and data faithfulness; The scope and meaning of LIME explanations depend on the kernel width and the notion of complexity [8, 37].

Solution: The suitability of an IML method cannot be evaluated with respect to one-fits-all interpretability but must be motivated and assessed with respect to well-defined interpretation goals. Similarly, practitioners must tailor the choice of the IML method and its respective hyperparameters to the interpretation context. This implies that these goals need to be clearly stated in a detailed manner *before* any analysis – which is still often not the case.

Open Issues: Since IML methods themselves are subject to interpretation, practitioners must be informed about which conclusions can or cannot be drawn given different choices of IML technique. In general, there are three aspects to be considered: (a) an intuitively understandable and plausible algorithmic construction of the IML method to achieve an explanation; (b) a clear mathematical axiomatization of interpretation goals and properties, which are linked by proofs and theoretical considerations to IML methods, and properties of models and data characteristics; (c) a practical translation for practitioners of the axioms from (b) in terms of what an IML method provides and what not, ideally with implementable guidelines and diagnostic checks for violated assumptions to guarantee correct interpretations. While (a) is nearly always given for any published method, much work remains for (b) and (c).

3 Bad Model Generalization

Pitfall: Under- or overfitting models can result in misleading interpretations with respect to the true feature effects and importance scores, as the model does not match the underlying data-generating process well [39]. Formally, most IML methods are designed to interpret the model instead of drawing inferences about

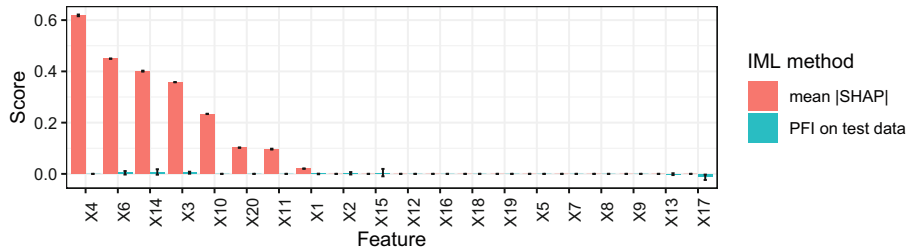


Fig. 2. Assuming one-fits-all interpretability. A default `xgboost` regression model that minimizes the mean squared error (MSE) was fitted on 20 independently and uniformly distributed features to predict another independent, uniformly sampled target. In this setting, predicting the (unconditional) mean $\mathbb{E}[Y]$ in a constant model is optimal. The learner overfits due to a small training data size. Mean marginal SHAP (red, error bars indicate 0.05 and 0.95 quantiles) exposes all mechanistically used features. In contrast, PFI on test data (blue, error bars indicate 0.05 and 0.95 quantiles) considers all features to be irrelevant, since no feature contributes to the generalization performance.

the data-generating process. In practice, however, the latter is often the goal of the analysis, and then an interpretation can only be as good as its underlying model. If a model approximates the data-generating process well enough, its interpretation should reveal insights into the underlying process.

Solution: In-sample evaluation (i.e. on training data) should not be used to assess the performance of ML models due to the risk of overfitting on the training data, which will lead to overly optimistic performance estimates. We must resort to out-of-sample validation based on resampling procedures such as hold-out for larger datasets or cross-validation, or even repeated cross-validation for small sample size scenarios. These resampling procedures are readily available in software [67,89], and well-studied in theory as well as practice [4,11,104], although rigorous analysis of cross-validation is still considered an open problem [103]. Nested resampling is necessary, when computational model selection and hyperparameter tuning are involved [10]. This is important, as the Bayes error for most practical situations is unknown, and we cannot make absolute statements about whether a model already optimally fits the data.

Figure 3 shows the mean squared errors for a simulated example on both training and test data for a support vector machine (SVM), a random forest, and a linear model. Additionally, PDPs for all models are displayed, which show to what extent each model’s effect estimates deviate from the ground truth. The linear model is unable to represent the non-linear relationship, which is reflected in a high error on both test and training data and the linear PDPs. In contrast, the random forest has a low training error but a much higher test error, which indicates overfitting. Also, the PDPs for the random forest display overfitting behavior, as the curves are quite noisy, especially at the lower and upper value

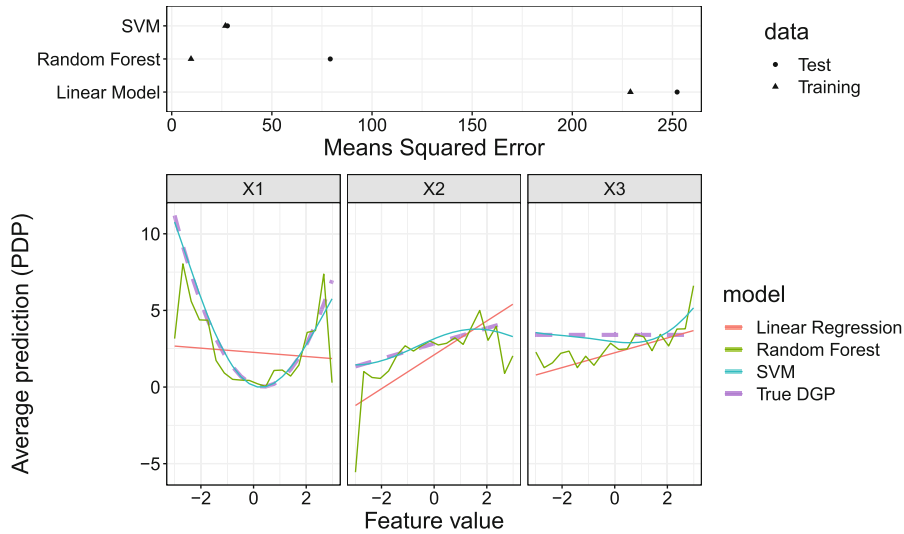


Fig. 3. Bad model generalization. **Top:** Performance estimates on training and test data for a linear regression model (underfitting), a random forest (overfitting) and a support vector machine with radial basis kernel (good fit). The three features are drawn from a uniform distribution, and the target was generated as $Y = X_1^2 + X_2 - 5X_1X_2 + \epsilon$, with $\epsilon \sim N(0, 5)$. **Bottom:** PDPs for the data-generating process (DGP) – which is the ground truth – and for the three models.

ranges of each feature. The SVM with both low training and test error comes closest to the true PDPs.

4 Unnecessary Use of Complex Models

Pitfall: A common mistake is to use an opaque, complex ML model when an interpretable model would have been sufficient, i.e. when the performance of interpretable models is only negligibly worse – or maybe the same or even better – than that of the ML model. Although model-agnostic methods can shed light on the behavior of complex ML models, inherently interpretable models still offer a higher degree of transparency [95] and considering them increases the chance of discovering the true data-generating function [23]. What constitutes an interpretable model is highly dependent on the situation and target audience, as even a linear model might be difficult to interpret when many features and interactions are involved.

It is commonly believed that complex ML models always outperform more interpretable models in terms of accuracy and should thus be preferred. However, there are several examples where interpretable models have proven to be serious competitors: More than 15 years ago, Hand [49] demonstrated that simple models often achieve more than 90% of the predictive power of potentially highly complex models across the UCI benchmark data repository and concluded that such

models often should be preferred due to their inherent interpretability; Makridakis et al. [79] systematically compared various ML models (including long-short-term-memory models and multi-layer neural networks) to statistical models (e.g. damped exponential smoothing and the Theta method) in time series forecasting tasks and found that the latter consistently show greater predictive accuracy; Kuhle et al. [65] found that random forests, gradient boosting and neural networks did not outperform logistic regression in predicting fetal growth abnormalities; Similarly, Wu et al. [120] have shown that a logistic regression model performs as well as AdaBoost and even better than an SVM in predicting heart disease from electronic health record data; Baesens et al. [7] showed that simple interpretable classifiers perform competitively for credit scoring, and in an update to the study the authors note that “the complexity and/or recency of a classifier are misleading indicators of its prediction performance” [71].

Solution: We recommend starting with simple, interpretable models such as linear regression models and decision trees. Generalized additive models (GAM) [50] can serve as a gradual transition between simple linear models and more complex machine learning models. GAMs have the desirable property that they can additively model smooth, non-linear effects and provide PDPs out-of-the-box, but without the potential pitfall of masking interactions (see Sect. 6). The additive model structure of a GAM is specified before fitting the model so that only the pre-specified feature or interaction effects are estimated. Interactions between features can be added manually or algorithmically (e.g. via a forward greedy search) [18]. GAMs can be fitted with component-wise boosting [99]. The boosting approach allows to smoothly increase model complexity, from sparse linear models to more complex GAMs with non-linear effects and interactions. This smooth transition provides insight into the tradeoffs between model simplicity and performance gains. Furthermore, component-wise boosting has an in-built feature selection mechanism as the model is build incrementally, which is especially useful in high-dimensional settings (see Sect. 9.1). The predictive performance of models of different complexity should be carefully measured and compared. Complex models should only be favored if the additional performance gain is both significant and relevant – a judgment call that the practitioner must ultimately make. Starting with simple models is considered best practice in data science, independent of the question of interpretability [23]. The comparison of predictive performance between model classes of different complexity can add further insights for interpretation.

Open Issues: Measures of model complexity allow quantifying the trade-off between complexity and performance and to automatically optimize for multiple objectives beyond performance. Some steps have been made towards quantifying model complexity, such as using functional decomposition and quantifying the complexity of the components [82] or measuring the stability of predictions [92]. However, further research is required, as there is no single perfect definition of interpretability, but rather multiple depending on the context [30, 95].

5 Ignoring Feature Dependence

5.1 Interpretation with Extrapolation

Pitfall: When features are dependent, perturbation-based IML methods such as PFI, PDP, LIME, and Shapley values extrapolate in areas where the model was trained with little or no training data, which can cause misleading interpretations [55]. This is especially true if the ML model relies on feature interactions [45] – which is often the case. Perturbations produce artificial data points that are used for model predictions, which in turn are aggregated to produce global or local interpretations [100]. Feature values can be perturbed by replacing original values with values from an equidistant grid of that feature, with permuted or randomly subsampled values [19], or with quantiles. We highlight two major issues: First, if features are dependent, all three perturbation approaches produce unrealistic data points, i.e. the new data points are located outside of the multivariate joint distribution of the data (see Fig. 4). Second, even if features are independent, using an equidistant grid can produce unrealistic values for the feature of interest. Consider a feature that follows a skewed distribution with outliers. An equidistant grid would generate many values between outliers and non-outliers. In contrast to the grid-based approach, the other two approaches maintain the marginal distribution of the feature of interest.

Both issues can result in misleading interpretations (illustrative examples are given in [55, 84]), since the model is evaluated in areas of the feature space with few or no observed real data points, where model uncertainty can be expected to be very high. This issue is aggravated if interpretation methods integrate over such points with the same weight and confidence as for much more realistic samples with high model confidence.

Solution: Before applying interpretation methods, practitioners should check for dependencies between features in the data, e.g. via descriptive statistics or measures of dependence (see Sect. 5.2). When it is unavoidable to include dependent features in the model (which is usually the case in ML scenarios), additional information regarding the strength and shape of the dependence structure should be provided. Sometimes, alternative interpretation methods can be used as a workaround or to provide additional information. Accumulated local effect plots (ALE) [3] can be applied when features are dependent, but can produce non-intuitive effect plots for simple linear models with interactions [45]. For other methods such as the PFI, conditional variants exist [17, 84, 107]. In the case of LIME, it was suggested to focus in sampling on realistic (i.e. close to the data manifold) [97] and relevant areas (e.g. close to the decision boundary) [69]. Note, however, that conditional interpretations are often different and should not be used as a substitute for unconditional interpretations (see Sect. 5.3). Furthermore, dependent features should not be interpreted separately but rather jointly. This can be achieved by visualizing e.g. a 2-dimensional ALE plot of two dependent features, which, admittedly, only works for very low-dimensional combinations. Especially in high-dimensional settings where dependent features

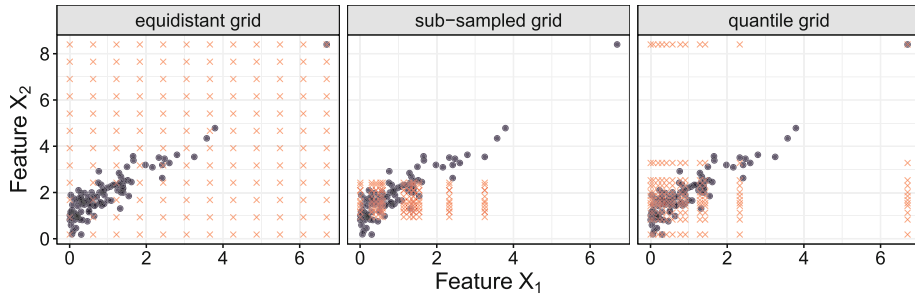


Fig. 4. Interpretation with extrapolation. Illustration of artificial data points generated by three different perturbation approaches. The black dots refer to observed data points and the red crosses to the artificial data points.

can be grouped in a meaningful way, grouped interpretation methods might be more reasonable (see Sect. 9.1).

We recommend using quantiles or randomly subsampled values over equidistant grids. By default, many implementations of interpretability methods use an equidistant grid to perturb feature values [41, 81, 89], although some also allow using user-defined values.

Open Issues: A comprehensive comparison of strategies addressing extrapolation and how they affect an interpretation method is currently missing. This also includes studying interpretation methods and their conditional variants when they are applied to data with different dependence structures.

5.2 Confusing Linear Correlation with General Dependence

Pitfall: Features with a Pearson correlation coefficient (PCC) close to zero can still be dependent and cause misleading model interpretations (see Fig. 5). While independence between two features implies that the PCC is zero, the converse is generally false. The PCC, which is often used to analyze dependence, only tracks linear correlations and has other shortcomings such as sensitivity to outliers [113]. Any type of dependence between features can have a strong impact on the interpretation of the results of IML methods (see Sect. 5.1). Thus, knowledge about the (possibly non-linear) dependencies between features is crucial for an informed use of IML methods.

Solution: Low-dimensional data can be visualized to detect dependence (e.g. scatter plots) [80]. For high-dimensional data, several other measures of dependence in addition to PCC can be used. If dependence is monotonic, Spearman’s rank correlation coefficient [72] can be a simple, robust alternative to PCC. For categorical or mixed features, separate dependence measures have been proposed, such as Kendall’s rank correlation coefficient for ordinal features, or the phi coefficient and Goodman & Kruskal’s lambda for nominal features [59].

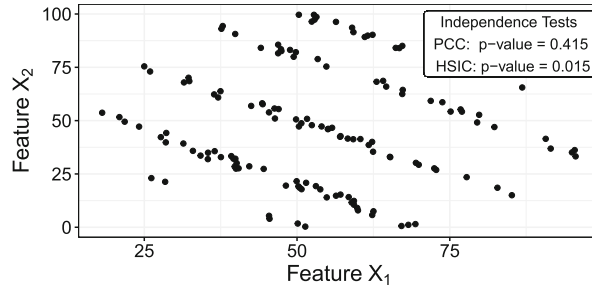


Fig. 5. Confusing linear correlation with dependence. Highly dependent features X_1 and X_2 that have a correlation close to zero. A test (H_0 : Features are independent) using Pearson correlation is not significant, but for HSIC, the H_0 -hypothesis gets rejected. Data from [80].

Studying non-linear dependencies is more difficult since a vast variety of possible associations have to be checked. Nevertheless, several non-linear association measures with sound statistical properties exist. Kernel-based measures, such as kernel canonical correlation analysis (KCCA) [6] or the Hilbert-Schmidt independence criterion (HSIC) [44], are commonly used. They have a solid theoretical foundation, are computationally feasible, and robust [113]. In addition, there are information-theoretical measures, such as (conditional) mutual information [24] or the maximal information coefficient (MIC) [93], that can however be difficult to estimate [9, 116]. Other important measures are e.g. the distance correlation [111], the randomized dependence coefficient (RDC) [74], or the alternating conditional expectations (ACE) algorithm [14]. In addition to using PCC, we recommend using at least one measure that detects non-linear dependencies (e.g. HSIC).

5.3 Misunderstanding Conditional Interpretation

Pitfall: Conditional variants of interpretation techniques avoid extrapolation but require a different interpretation. Interpretation methods that perturb features independently of others will extrapolate under dependent features but provide insight into the model’s mechanism [56, 61]. Therefore, these methods are said to be true to the model but not true to the data [21].

For feature effect methods such as the PDP, the plot can be interpreted as the isolated, average effect the feature has on the prediction. For the PFI, the importance can be interpreted as the drop in performance when the feature’s information is “destroyed” (by perturbing it). Marginal SHAP value functions [78] quantify a feature’s contribution to a specific prediction, and marginal SAGE value functions [25] quantify a feature’s contribution to the overall prediction performance. All the aforementioned methods extrapolate under dependent features (see also Sect. 5.1), but satisfy sensitivity, i.e. are zero if a feature is not used by the model [25, 56, 61, 110].

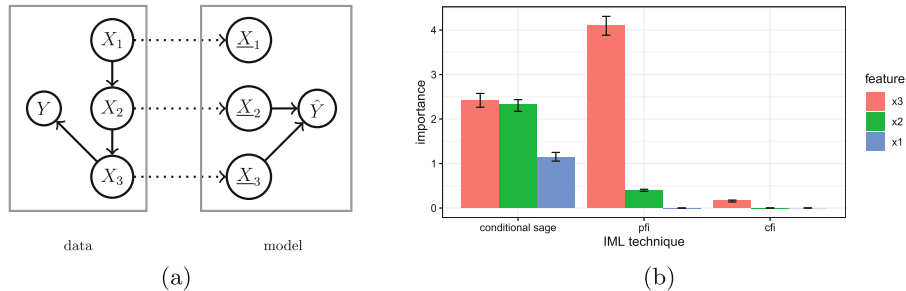


Fig. 6. Misunderstanding conditional interpretation. A linear model was fitted on the data-generating process modeled using a linear Gaussian structural causal model. The entailed directed acyclic graph is depicted on the left. For illustrative purposes, the original model coefficients were updated such that not only feature X_3 , but also feature X_2 is used by the model. PFI on test data considers both X_3 and X_2 to be relevant. In contrast, conditional feature importance variants either only consider X_3 to be relevant (CFI) or consider all features to be relevant (conditional SAGE value function).

Conditional variants of these interpretation methods do not replace feature values independently of other features, but in such a way that they conform to the conditional distribution. This changes the interpretation as the effects of all dependent features become entangled. Depending on the method, conditional sampling leads to a more or less restrictive notion of relevance.

For example, for dependent features, the Conditional Feature Importance (CFI) [17, 84, 107, 117] answers the question: “How much does the model performance drop if we permute a feature, *but given that we know the values of the other features?*” [63, 84, 107].¹ Two highly dependent features might be individually important (based on the unconditional PFI), but have a very low conditional importance score because the information of one feature is contained in the other and vice versa.

In contrast, the conditional variant of PDP, called marginal plot or M-plot [3], violates sensitivity, i.e. may even show an effect for features that are not used by the model. This is because for M-plots, the feature of interest is not sampled conditionally on the remaining features, but rather the remaining features are sampled conditionally on the feature of interest. As a consequence, the distribution of dependent covariates varies with the value of the feature of interest. Similarly, conditional SAGE and conditional SHAP value functions sample the remaining features conditional on the feature of interest and therefore violate sensitivity [25, 56, 61, 109].

We demonstrate the difference between PFI, CFI, and conditional SAGE value functions on a simulated example (Fig. 6) where the data-generating mech-

¹ While for CFI the conditional independence of the feature of interest X_j with the target Y given the remaining features X_{-j} ($Y \perp X_j | X_{-j}$) is already a sufficient condition for zero importance, the corresponding PFI may still be nonzero [63].

anism is known. While PFI only considers features to be relevant if they are actually used by the model, SAGE value functions may also consider a feature to be important that is not directly used by the model if it contains information that the model exploits. CFI only considers a feature to be relevant if it is both mechanistically used by the model and contributes unique information about Y .

Solution: When features are highly dependent and conditional effects and importance scores are used, the practitioner must be aware of the distinct interpretation. Recent work formalizes the implications of marginal and conditional interpretation techniques [21, 25, 56, 61, 63]. While marginal methods provide insight into the model’s mechanism but are not true to the data, their conditional variants are not true to the model but provide insight into the associations in the data.

If joint insight into model and data is required, designated methods must be used. ALE plots [3] provide interval-wise unconditional interpretations that are true to the data. They have been criticized to produce non-intuitive results for certain data-generating mechanisms [45]. Molnar et al. [84] propose a subgroup-based conditional sampling technique that allows for group-wise marginal interpretations that are true to model and data and that can be applied to feature importance and feature effects methods such as conditional PDPs and CFI. For feature importance, the DEDACT framework [61] allows to decompose conditional importance measures such as SAGE value functions into their marginal contributions and vice versa, thereby allowing global insight into both: the sources of prediction-relevant information in the data as well as into the feature pathways by which the information enters the model.

Open Issues: The quality of conditional IML techniques depends on the goodness of the conditional sampler. Especially in continuous, high-dimensional settings, conditional sampling is challenging. More research on the robustness of interpretation techniques regarding the quality of the sample is required.

6 Misleading Interpretations Due to Feature Interactions

6.1 Misleading Feature Effects Due to Aggregation

Pitfall: Global interpretation methods, such as PDP or ALE plots, visualize the average effect of a feature on a model’s prediction. However, they can produce misleading interpretations when features interact. Figure 7 A and B show the marginal effect of features X_1 and X_2 of the below-stated simulation example. While the PDP of the non-interacting feature X_1 seems to capture the true underlying effect of X_1 on the target quite well (A), the global aggregated effect of the interacting feature X_2 (B) shows almost no influence on the target, although an effect is clearly there by construction.

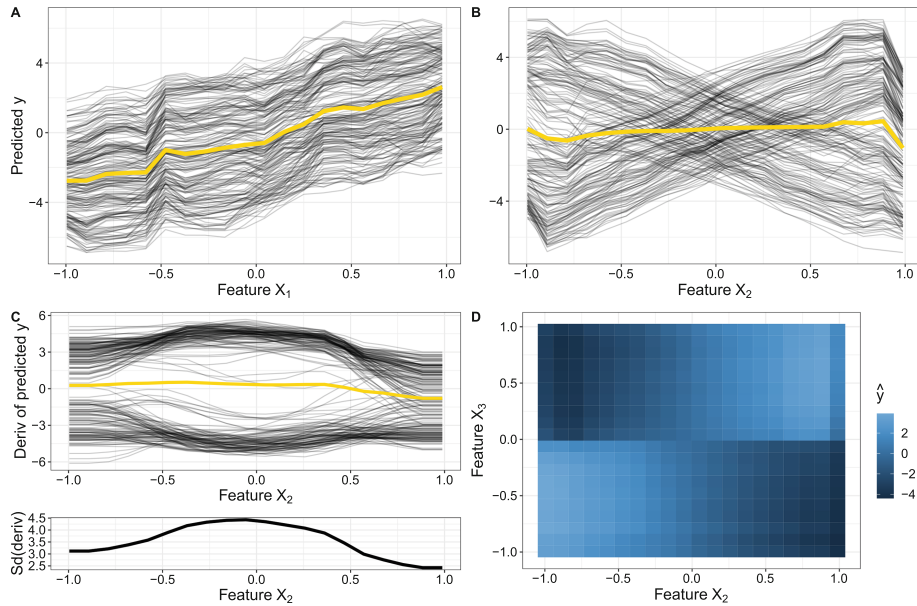


Fig. 7. Misleading effect due to interactions. Simulation example with interactions: $Y = 3X_1 - 6X_2 + 12X_2\mathbb{1}_{(X_3 \geq 0)} + \epsilon$ with $X_1, X_2, X_3 \stackrel{i.i.d.}{\sim} U[-1, 1]$ and $\epsilon \stackrel{i.i.d.}{\sim} N(0, 0.3)$. A random forest with 500 trees is fitted on 1000 observations. Effects are calculated on 200 randomly sampled (training) observations. **A**, **B**: PDP (yellow) and ICE curves of X_1 and X_2 ; **C**: Derivative ICE curves and their standard deviation of X_2 ; **D**: 2-dimensional PDP of X_2 and X_3 .

Solution: For the PDP, we recommend to additionally consider the corresponding ICE curves [38]. While PDP and ALE average out interaction effects, ICE curves directly show the heterogeneity between individual predictions. Figure 7 A illustrates that the individual marginal effect curves all follow an upward trend with only small variations. Hence, by aggregating these ICE curves to a global marginal effect curve such as the PDP, we do not lose much information. However, when the regarded feature interacts with other features, such as feature X_2 with feature X_3 in this example, then marginal effect curves of different observations might not show similar effects on the target. Hence, ICE curves become very heterogeneous, as shown in Fig. 7 B. In this case, the influence of feature X_2 is not well represented by the global average marginal effect. Particularly for continuous interactions where ICE curves start at different intercepts, we recommend the use of derivative or centered ICE curves, which eliminate differences in intercepts and leave only differences due to interactions [38]. Derivative ICE curves also point out the regions of highest interaction with other features. For example, Fig. 7 C indicates that predictions for X_2 taking values close to 0 strongly depend on other features' values. While these methods show that interactions are present with regards to the feature of interest but do not reveal other

features with which it interacts, the 2-dimensional PDP or ALE plot are options to visualize 2-way interaction effects. The 2-dimensional PDP in Fig. 7 D shows that predictions with regards to feature X_2 highly depend on the feature values of feature X_3 .

Other methods that aim to gain more insights into these visualizations are based on clustering homogeneous ICE curves, such as visual interaction effects (VINE) [16] or [122]. As an example, in Fig. 7 B, it would be more meaningful to average over the upward and downward proceeding ICE curves separately and hence show that the average influence of feature X_2 on the target depends on an interacting feature (here: X_3). Work by Zon et al. [125] followed a similar idea by proposing an interactive visualization tool to group Shapley values with regards to interacting features that need to be defined by the user.

Open Issues: The introduced visualization methods are not able to illustrate the type of the underlying interaction and most of them are also not applicable to higher-order interactions.

6.2 Failing to Separate Main from Interaction Effects

Pitfall: Many interpretation methods that quantify a feature’s importance or effect cannot separate an interaction from main effects. The PFI, for example, includes both the importance of a feature and the importance of all its interactions with other features [19]. Also local explanation methods such as LIME and Shapley values only provide additive explanations without separation of main effects and interactions [40].

Solution: Functional ANOVA introduced by [53] is probably the most popular approach to decompose the joint distribution into main and interaction effects. Using the same idea, the H-Statistic [35] quantifies the interaction strength between two features or between one feature and all others by decomposing the 2-dimensional PDP into its univariate components. The H-Statistic is based on the fact that, in the case of non-interacting features, the 2-dimensional partial dependence function equals the sum of the two underlying univariate partial dependence functions. Another similar interaction score based on partial dependencies is defined by [42]. Instead of decomposing the partial dependence function, [87] uses the predictive performance to measure interaction strength. Based on Shapley values, Lundberg et al. [77] proposed SHAP interaction values, and Casalicchio et al. [19] proposed a fair attribution of the importance of interactions to the individual features.

Furthermore, Hooker [54] considers dependent features and decomposes the predictions in main and interaction effects. A way to identify higher-order interactions is shown in [53].

Open Issues: Most methods that quantify interactions are not able to identify higher-order interactions and interactions of dependent features. Furthermore,

the presented solutions usually lack automatic detection and ranking of all interactions of a model. Identifying a suitable shape or form of the modeled interaction is not straightforward as interactions can be very different and complex, e.g., they can be a simple product of features (multiplicative interaction) or can have a complex joint non-linear effect such as smooth spline surface.

7 Ignoring Model and Approximation Uncertainty

Pitfall: Many interpretation methods only provide a mean estimate but do not quantify uncertainty. Both the model training and the computation of interpretation are subject to uncertainty. The model is trained on (random) data, and therefore should be regarded as a random variable. Similarly, LIME’s surrogate model relies on perturbed and reweighted samples of the data to approximate the prediction function locally [94]. Other interpretation methods are often defined in terms of expectations over the data (PFI, PDP, Shapley values, ...), but are approximated using Monte Carlo integration. Ignoring uncertainty can result in the interpretation of noise and non-robust results. The true effect of a feature may be flat, but – purely by chance, especially on smaller datasets – the Shapley value might show an effect. This effect could cancel out once averaged over multiple model fits.

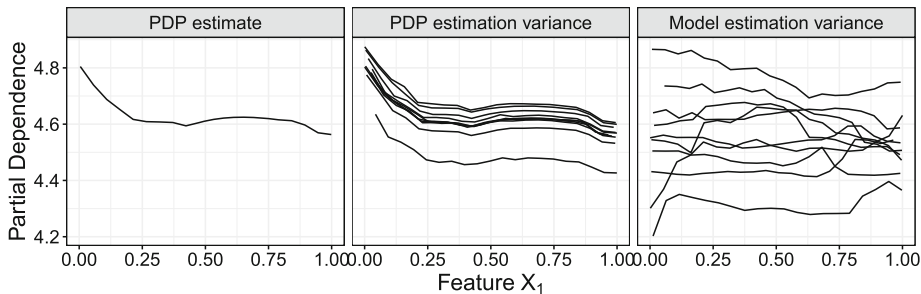


Fig. 8. Ignoring model and approximation uncertainty. PDP for X_1 with $Y = 0 \cdot X_1 + \sum_{j=2}^{10} X_j + \epsilon_i$ with $X_1, \dots, X_{10} \sim U[0, 1]$ and $\epsilon_i \sim N(0, 0.9)$. **Left:** PDP for X_1 of a random forest trained on 100 data points. **Middle:** Multiple PDPs (10x) for the model from left plots, but with different samples (each $n=100$) for PDP estimation. **Right:** Repeated (10x) data samples of $n=100$ and newly fitted random forest.

Figure 8 shows that a single PDP (first plot) can be misleading because it does not show the variance due to PDP estimation (second plot) and model fitting (third plot). If we are not interested in learning about a specific model, but rather about the relationship between feature X_1 and the target (in this case), we should consider the model variance.

Solution: By repeatedly computing PDP and PFI with a given model, but with different permutations or bootstrap samples, the uncertainty of the estimate can be quantified, for example in the form of confidence intervals. For PFI, frameworks for confidence intervals and hypothesis tests exist [2, 117], but they assume a fixed model. If the practitioner wants to condition the analysis on the modeling process and capture the process’ variance instead of conditioning on a fixed model, PDP and PFI should be computed on multiple model fits [83].

Open Issues: While Moosbauer et al. [85] derived confidence bands for PDPs for probabilistic ML models that cover the model’s uncertainty, a general model-agnostic uncertainty measure for feature effect methods such as ALE [3] and PDP [36] has (to the best of our knowledge) not been introduced yet.

8 Ignoring the Rashomon Effect

Pitfall: Sometimes different models explain the data-generating process equally well, but contradict each other. This phenomenon is called the Rashomon effect, named after the movie “Rashomon” from the year 1950. Breiman formalized it for predictive models in 2001 [13]: Different prediction models might perform equally well (Rashomon set), but construct the prediction function in a different way (e.g. relying on different features). This can result in conflicting interpretations and conclusions about the data. Even small differences in the training data can cause one model to be preferred over another.

For example, Dong and Rudin [29] identified a Rashomon set of equally well performing models for the COMPAS dataset. They showed that the models differed greatly in the importance they put on certain features. Specifically, if criminal history was identified as less important, race was more important and vice versa. Cherry-picking one model and its underlying explanation might not be sufficient to draw conclusions about the data-generating process. As Hancox-Li [48] states “just because race happens to be an unimportant variable in that one explanation does not mean that it is objectively an unimportant variable”.

The Rashomon effect can also occur at the level of the interpretation method itself. Differing hyperparameters or interpretation goals can be one reason (see Sect. 2). But even if the hyperparameters are fixed, we could still obtain contradicting explanations by an interpretation method, e.g., due to a different data sample or initial seed.

A concrete example of the Rashomon effect is counterfactual explanations. Different counterfactuals may all alter the prediction in the desired way, but point to different feature changes required for that change. If a person is deemed uncreditworthy, one corresponding counterfactual explaining this decision may point to a scenario in which the person had asked for a shorter loan duration and amount, while another counterfactual may point to a scenario in which the person had a higher income and more stable job. Focusing on only one counterfactual explanation in such cases strongly limits the possible epistemic access.

Solution: If multiple, equally good models exist, their interpretations should be compared. Variable importance clouds [29] is a method for exploring variable importance scores for equally good models within one model class. If the interpretations are in conflict, conclusions must be drawn carefully. Domain experts or further constraints (e.g. fairness or sparsity) could help to pick a suitable model. Semenova et al. [102] also hypothesized that a large Rashomon set could contain simpler or more interpretable models, which should be preferred according to Sect. 4.

In the case of counterfactual explanations, multiple, equally good explanations exist. Here, methods that return a set of explanations rather than a single one should be used – for example, the method by Dandl et al. [26] or Mothilal et al. [86].

Open Issues: Numerous very different counterfactual explanations are overwhelming for users. Methods for aggregating or combining explanations are still a matter of future research.

9 Failure to Scale to High-Dimensional Settings

9.1 Human-Intelligibility of High-Dimensional IML Output

Pitfall: Applying IML methods naively to high-dimensional datasets (e.g. visualizing feature effects or computing importance scores on feature level) leads to an overwhelming and high-dimensional IML output, which impedes human analysis. Especially interpretation methods that are based on visualizations make it difficult for practitioners in high-dimensional settings to focus on the most important insights.

Solution: A natural approach is to reduce the dimensionality before applying any IML methods. Whether this facilitates understanding or not depends on the possible semantic interpretability of the resulting, reduced feature space – as features can either be selected or dimensionality can be reduced by linear or non-linear transformations. Assuming that users would like to interpret in the original feature space, many feature selection techniques can be used [46], resulting in much sparser and consequently easier to interpret models. Wrapper selection approaches are model-agnostic and algorithms like greedy forward selection or subset selection procedures [5, 60], which start from an empty model and iteratively add relevant (subsets of) features if needed, even allow to measure the relevance of features for predictive performance. An alternative is to directly use models that implicitly perform feature selection such as LASSO [112] or component-wise boosting [99] as they can produce sparse models with fewer features. In the case of LIME or other interpretation methods based on surrogate models, the aforementioned techniques could be applied to the surrogate model.

When features can be meaningfully grouped in a data-driven or knowledge-driven way [51], applying IML methods directly to grouped features instead of

single features is usually more time-efficient to compute and often leads to more appropriate interpretations. Examples where features can naturally be grouped include the grouping of sensor data [20], time-lagged features [75], or one-hot-encoded categorical features and interaction terms [43]. Before a model is fitted, groupings could already be exploited for dimensionality reduction, for example by selecting groups of features by the group LASSO [121].

For model interpretation, various papers extended feature importance methods from single features to groups of features [5, 43, 114, 119]. In the case of grouped PFI, this means that we perturb the entire group of features at once and measure the performance drop compared to the unperturbed dataset. Compared to standard PFI, the grouped PFI does not break the association to the other features of the group, but to features of other groups and the target. This is especially useful when features within the same group are highly correlated (e.g. time-lagged features), but between-group dependencies are rather low. Hence, this might also be a possible solution for the extrapolation pitfall described in Sect. 5.1.

We consider the PhoneStudy in [106] as an illustration. The PhoneStudy dataset contains 1821 features to analyze the link between human behavior based on smartphone data and participants' personalities. Interpreting the results in this use case seems to be challenging since features were dependent and single feature effects were either small or non-linear [106]. The features have been grouped in behavior-specific categories such as app-usage, music consumption, or overall phone usage. Au et al. [5] calculated various grouped importance scores on the feature groups to measure their influence on a specific personality trait (e.g. conscientiousness). Furthermore, the authors applied a greedy forward subset selection procedure via repeated subsampling on the feature groups and showed that combining app-usage features and overall phone usage features were most of the times sufficient for the given prediction task.

Open Issues: The quality of a grouping-based interpretation strongly depends on the human intelligibility and meaningfulness of the grouping. If the grouping structure is not naturally given, then data-driven methods can be used. However, if feature groups are not meaningful (e.g. if they cannot be described by a super-feature such as app-usage), then subsequent interpretations of these groups are purposeless. One solution could be to combine feature selection strategies with interpretation methods. For example, LIME's surrogate model could be a LASSO model. However, beyond surrogate models, the integration of feature selection strategies remains an open issue that requires further research.

Existing research on grouped interpretation methods mainly focused on quantifying grouped feature importance, but the question of "how a group of features influences a model's prediction" remains almost unanswered. Only recently, [5, 15, 101] attempted to answer this question by using dimension-reduction techniques (such as PCA) before applying the interpretation method. However, this is also a matter of further research.

9.2 Computational Effort

Pitfall: Some interpretation methods do not scale linearly with the number of features. For example, for the computation of exact Shapley values the number of possible coalitions [25,78], or for a (full) functional ANOVA decomposition the number of components (main effects plus all interactions) scales with $\mathcal{O}(2^p)$ [54].²

Solution: For the functional ANOVA, a common solution is to keep the analysis to the main effects and selected 2-way interactions (similar for PDP and ALE). Interesting 2-way interactions can be selected by another method such as the H-statistic [35]. However, the selection of 2-way interactions requires additional computational effort. Interaction strength usually decreases quickly with increasing interaction size, and one should only consider d -way interactions when all their $(d-1)$ -way interactions were significant [53]. For Shapley-based methods, an efficient approximation exists that is based on randomly sampling and evaluating feature orderings until the estimates converge. The variance of the estimates reduces in $\mathcal{O}(\frac{1}{m})$, where m is the number of evaluated orderings [25,78].

9.3 Ignoring Multiple Comparison Problem

Pitfall: Simultaneously testing the importance of multiple features will result in false-positive interpretations if the multiple comparisons problem (MCP) is ignored. The MCP is well known in significance tests for linear models and exists similarly in testing for feature importance in ML. For example, suppose we simultaneously test the importance of 50 features (with the H_0 -hypothesis of zero importance) at the significance level $\alpha = 0.05$. Even if all features are unimportant, the probability of observing that at least one feature is significantly important is $1 - \mathbb{P}(\text{'no feature important'}) = 1 - (1 - 0.05)^{50} \approx 0.923$. Multiple comparisons become even more problematic the higher the dimension of the dataset.

Solution: Methods such as Model-X knockoffs [17] directly control for the false discovery rate (FDR). For all other methods that provide p-values or confidence intervals, such as PIMP (Permutation IMPortance) [2], which is a testing approach for PFI, MCP is often ignored in practice to the best of our knowledge, with some exceptions [105,117]. One of the most popular MCP adjustment methods is the Bonferroni correction [31], which rejects a null hypothesis if its p-value is smaller than α/p , with p as the number of tests. It has the disadvantage that it increases the probability of false negatives [90]. Since MCP is well known in statistics, we refer the practitioner to [28] for an overview and discussion of alternative adjustment methods, such as the Bonferroni-Holm method [52].

² Similar to the PDP or ALE plots, the functional ANOVA components describe individual feature effects and interactions.

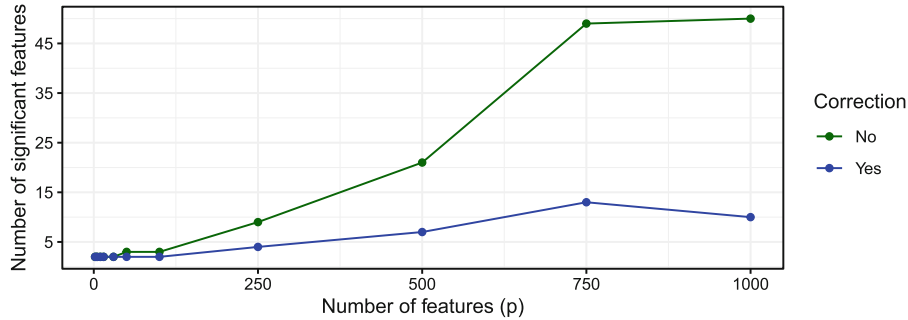


Fig. 9. Failure to scale to high-dimensional settings. Comparison of the number of features with significant importance - once with and once without Bonferroni-corrected significance levels for a varying number of added noise variables. Datasets were sampled from $Y = 2X_1 + 2X_2^2 + \epsilon$ with $X_1, X_2, \epsilon \sim N(0, 1)$. $X_3, X_4, \dots, X_p \sim N(0, 1)$ are additional noise variables with p ranging between 2 and 1000. For each p , we sampled two datasets from this data-generating process – one to train a random forest with 500 trees on and one to test whether feature importances differed from 0 using PIMP. In all experiments, X_1 and X_2 were correctly identified as important.

As an example, in Fig. 9 we compare the number of features with significant importance measured by PIMP once with and once without Bonferroni-adjusted significance levels ($\alpha = 0.05$ vs. $\alpha = 0.05/p$). Without correcting for multiple comparisons, the number of features mistakenly evaluated as important grows considerably with increasing dimension, whereas Bonferroni correction results in only a modest increase.

10 Unjustified Causal Interpretation

Pitfall: Practitioners are often interested in causal insights into the underlying data-generating mechanisms, which IML methods do not generally provide. Common causal questions include the identification of causes and effects, predicting the effects of interventions, and answering counterfactual questions [88]. For example, a medical researcher might want to identify risk factors or predict average and individual treatment effects [66]. In search of answers, a researcher can therefore be tempted to interpret the result of IML methods from a causal perspective.

However, a causal interpretation of predictive models is often not possible. Standard supervised ML models are not designed to model causal relationships but to merely exploit associations. A model may therefore rely on causes and effects of the target variable as well as on variables that help to reconstruct unobserved influences on Y , e.g. causes of effects [118]. Consequently, the question of whether a variable is relevant to a predictive model (indicated e.g. by $\text{PFI} > 0$) does not directly indicate whether a variable is a cause, an effect, or does not stand in any causal relation to the target variable. Furthermore,

even if a model would rely solely on direct causes for the prediction, the causal structure between features must be taken into account. Intervening on a variable in the real world may affect not only Y but also other variables in the feature set. Without assumptions about the underlying causal structure, IML methods cannot account for these adaptations and guide action [58, 62].

As an example, we constructed a dataset by sampling from a structural causal model (SCM), for which the corresponding causal graph is depicted in Fig. 10. All relationships are linear Gaussian with variance 1 and coefficients 1. For a linear model fitted on the dataset, all features were considered to be relevant based on the model coefficients ($\hat{y} = 0.329x_1 + 0.323x_2 - 0.327x_3 + 0.342x_4 + 0.334x_5$, $R^2 = 0.943$), although x_3 , x_4 and x_5 do not cause Y .

Solution: The practitioner must carefully assess whether sufficient assumptions can be made about the underlying data-generating process, the learned model, and the interpretation technique. If these assumptions are met, a causal interpretation may be possible. The PDP between a feature and the target can be interpreted as the respective average causal effect if the model performs well and the set of remaining variables is a valid adjustment set [123]. When it is known whether a model is deployed in a causal or anti-causal setting – i.e. whether the model attempts to predict an effect from its causes or the other way round – a partial identification of the causal roles based on feature relevance is possible (under strong and non-testable assumptions) [118]. Designated tools and approaches are available for causal discovery and inference [91].

Open Issues: The challenge of causal discovery and inference remains an open key issue in the field of ML. Careful research is required to make explicit under which assumptions what insight about the underlying data-generating mechanism can be gained by interpreting an ML model.

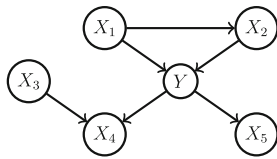


Fig. 10. Causal graph

11 Discussion

In this paper, we have reviewed numerous pitfalls of local and global model-agnostic interpretation techniques, e.g. in the case of bad model generalization, dependent features, interactions between features, or causal interpretations. We have not attempted to provide an exhaustive list of all potential pitfalls in ML

model interpretation, but have instead focused on common pitfalls that apply to various model-agnostic IML methods and pose a particularly high risk.

We have omitted pitfalls that are more specific to one IML method type: For local methods, the vague notions of neighborhood and distance can lead to misinterpretations [68, 69], and common distance metrics (such as the Euclidean distance) are prone to the curse of dimensionality [1]; Surrogate methods such as LIME may not be entirely faithful to the original model they replace in interpretation. Moreover, we have not addressed pitfalls associated with certain data types (like the definition of superpixels in image data [98]), nor those related to human cognitive biases (e.g. the illusion of model understanding [22]).

Many pitfalls in the paper are strongly linked with axioms that encode desiderata of model interpretation. For example, pitfall Sect. 5.3 (misunderstanding conditional interpretations) is related to violations of sensitivity [56, 110]. As such, axioms can help to make the strengths and limitations of methods explicit. Therefore, we encourage an axiomatic evaluation of interpretation methods.

We hope to promote a more cautious approach when interpreting ML models in practice, to point practitioners to already (partially) available solutions, and to stimulate further research on these issues. The stakes are high: ML algorithms are increasingly used for socially relevant decisions, and model interpretations play an important role in every empirical science. Therefore, we believe that users can benefit from concrete guidance on properties, dangers, and problems of IML techniques – especially as the field is advancing at high speed. We need to strive towards a recommended, well-understood set of tools, which will in turn require much more careful research. This especially concerns the meta-issues of comparisons of IML techniques, IML diagnostic tools to warn against misleading interpretations, and tools for analyzing multiple dependent or interacting features.

References

1. Aggarwal, C.C., Hinneburg, A., Keim, D.A.: On the surprising behavior of distance metrics in high dimensional space. In: Van den Bussche, J., Vianu, V. (eds.) ICDT 2001. LNCS, vol. 1973, pp. 420–434. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44503-X_27
2. Altmann, A., Tološi, L., Sander, O., Lengauer, T.: Permutation importance: a corrected feature importance measure. *Bioinformatics* **26**(10), 1340–1347 (2010). <https://doi.org/10.1093/bioinformatics/btq134>
3. Apley, D.W., Zhu, J.: Visualizing the effects of predictor variables in black box supervised learning models. *J. R. Stat. Soc.: Ser. B (Stat. Methodol.)* **82**(4), 1059–1086 (2020). <https://doi.org/10.1111/rssb.12377>
4. Arlot, S., Celisse, A.: A survey of cross-validation procedures for model selection. *Statist. Surv.* **4**, 40–79 (2010). <https://doi.org/10.1214/09-SS054>
5. Au, Q., Herbinger, J., Stachl, C., Bischl, B., Casalicchio, G.: Grouped feature importance and combined features effect plot. arXiv preprint [arXiv:2104.11688](https://arxiv.org/abs/2104.11688) (2021)
6. Bach, F.R., Jordan, M.I.: Kernel independent component analysis. *J. Mach. Learn. Res.* **3**(Jul), 1–48 (2002)

7. Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., Vanthienen, J.: Benchmarking state-of-the-art classification algorithms for credit scoring. *J. Oper. Res. Soc.* **54**(6), 627–635 (2003). <https://doi.org/10.1057/palgrave.jors.2601545>
8. Bansal, N., Agarwal, C., Nguyen, A.: SAM: the sensitivity of attribution methods to hyperparameters. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8673–8683 (2020)
9. Belghazi, M.I., et al.: Mutual information neural estimation. In: International Conference on Machine Learning, pp. 531–540 (2018)
10. Bischl, B., et al.: Hyperparameter optimization: foundations, algorithms, best practices and open challenges. arXiv preprint [arXiv:2107.05847](https://arxiv.org/abs/2107.05847) (2021)
11. Bischl, B., Mersmann, O., Trautmann, H., Weihs, C.: Resampling methods for meta-model validation with recommendations for evolutionary computation. *Evol. Comput.* **20**(2), 249–275 (2012). <https://doi.org/10.1162/EVCO.a.00069>
12. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001). <https://doi.org/10.1023/A:1010933404324>
13. Breiman, L.: Statistical modeling: the two cultures (with comments and a rejoinder by the author). *Stat. Sci.* **16**(3), 199–231 (2001). <https://doi.org/10.1214/ss/1009213726>
14. Breiman, L., Friedman, J.H.: Estimating optimal transformations for multiple regression and correlation. *J. Am. Stat. Assoc.* **80**(391), 580–598 (1985). <https://doi.org/10.1080/01621459.1985.10478157>
15. Brenning, A.: Transforming feature space to interpret machine learning models. [arXiv:2104.04295](https://arxiv.org/abs/2104.04295) (2021)
16. Britton, M.: Vine: visualizing statistical interactions in black box models. arXiv preprint [arXiv:1904.00561](https://arxiv.org/abs/1904.00561) (2019)
17. Candès, E., Fan, Y., Janson, L., Lv, J.: Panning for gold: ‘model-x’ knockoffs for high dimensional controlled variable selection. *J. R. Stat. Soc.: Ser. B (Stat. Methodol.)* **80**(3), 551–577 (2018). <https://doi.org/10.1111/rssb.12265>
18. Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., Elhadad, N.: Intelligible models for healthcare: predicting pneumonia risk and hospital 30-day readmission. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1721–1730 (2015). <https://doi.org/10.1145/2783258.2788613>
19. Casalicchio, G., Molnar, C., Bischl, B.: Visualizing the feature importance for black box models. In: Berlingerio, M., Bonchi, F., Gärtner, T., Hurley, N., Ifrim, G. (eds.) *ECML PKDD 2018. LNCS (LNAI)*, vol. 11051, pp. 655–670. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-10925-7_40
20. Chakraborty, D., Pal, N.R.: Selecting useful groups of features in a connectionist framework. *IEEE Trans. Neural Netw.* **19**(3), 381–396 (2008). <https://doi.org/10.1109/TNN.2007.910730>
21. Chen, H., Janizek, J.D., Lundberg, S., Lee, S.I.: True to the model or true to the data? arXiv preprint [arXiv:2006.16234](https://arxiv.org/abs/2006.16234) (2020)
22. Chromik, M., Eiband, M., Buchner, F., Krüger, A., Butz, A.: I think I get your point, AI! the illusion of explanatory depth in explainable AI. In: 26th International Conference on Intelligent User Interfaces, IUI 2021, pp. 307–317. Association for Computing Machinery, New York (2021). <https://doi.org/10.1145/3397481.3450644>
23. Claeskens, G., Hjort, N.L., et al.: *Model Selection and Model Averaging*. Cambridge Books (2008). <https://doi.org/10.1017/CBO9780511790485>

24. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley (2012). <https://doi.org/10.1002/047174882X>
25. Covert, I., Lundberg, S.M., Lee, S.I.: Understanding global feature contributions with additive importance measures. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems, vol. 33, pp. 17212–17223. Curran Associates, Inc. (2020)
26. Dandl, S., Molnar, C., Binder, M., Bischl, B.: Multi-objective counterfactual explanations. In: Bäck, T., et al. (eds.) PPSN 2020. LNCS, vol. 12269, pp. 448–469. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58112-1_31
27. Das, A., Rad, P.: Opportunities and challenges in explainable artificial intelligence (XAI): a survey. arXiv preprint [arXiv:2006.11371](https://arxiv.org/abs/2006.11371) (2020)
28. Dickhaus, T.: Simultaneous Statistical Inference. Springer, Heidelberg (2014). <https://doi.org/10.1007/978-3-642-45182-9>
29. Dong, J., Rudin, C.: Exploring the cloud of variable importance for the set of all good models. Nat. Mach. Intell. **2**(12), 810–824 (2020). <https://doi.org/10.1038/s42256-020-00264-0>
30. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning. arXiv preprint [arXiv:1702.08608](https://arxiv.org/abs/1702.08608) (2017)
31. Dunn, O.J.: Multiple comparisons among means. J. Am. Stat. Assoc. **56**(293), 52–64 (1961). <https://doi.org/10.1080/01621459.1961.10482090>
32. Fernández-Delgado, M., Cernadas, E., Barro, S., Amorim, D.: Do we need hundreds of classifiers to solve real world classification problems. J. Mach. Learn. Res. **15**(1), 3133–3181 (2014). <https://doi.org/10.5555/2627435.2697065>
33. Fisher, A., Rudin, C., Dominici, F.: All models are wrong, but many are useful: learning a variable’s importance by studying an entire class of prediction models simultaneously. J. Mach. Learn. Res. **20**(177), 1–81 (2019)
34. Freiesleben, T.: Counterfactual explanations & adversarial examples-common grounds, essential differences, and potential transfers. arXiv preprint [arXiv:2009.05487](https://arxiv.org/abs/2009.05487) (2020)
35. Friedman, J.H., Popescu, B.E.: Predictive learning via rule ensembles. Ann. Appl. Stat. **2**(3), 916–954 (2008). <https://doi.org/10.1214/07-AOAS148>
36. Friedman, J.H., et al.: Multivariate adaptive regression splines. Ann. Stat. **19**(1), 1–67 (1991). <https://doi.org/10.1214/aos/1176347963>
37. Garreau, D., von Luxburg, U.: Looking deeper into tabular lime. arXiv preprint [arXiv:2008.11092](https://arxiv.org/abs/2008.11092) (2020)
38. Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E.: Peeking inside the black box: visualizing statistical learning with plots of individual conditional expectation. J. Comput. Graph. Stat. **24**(1), 44–65 (2015). <https://doi.org/10.1080/10618600.2014.907095>
39. Good, P.I., Hardin, J.W.: Common Errors in Statistics (and How to Avoid Them). Wiley (2012). <https://doi.org/10.1002/9781118360125>
40. Gosiewska, A., Biecek, P.: Do not trust additive explanations. arXiv preprint [arXiv:1903.11420](https://arxiv.org/abs/1903.11420) (2019)
41. Greenwell, B.M.: PDP: an R package for constructing partial dependence plots. R J. **9**(1), 421–436 (2017). <https://doi.org/10.32614/RJ-2017-016>
42. Greenwell, B.M., Boehmke, B.C., McCarthy, A.J.: A simple and effective model-based variable importance measure. [arXiv:1805.04755](https://arxiv.org/abs/1805.04755) (2018)
43. Gregorutti, B., Michel, B., Saint-Pierre, P.: Grouped variable importance with random forests and application to multiple functional data analysis. Comput. Stat. Data Anal. **90**, 15–35 (2015). <https://doi.org/10.1016/j.csda.2015.04.002>

44. Gretton, A., Bousquet, O., Smola, A., Schölkopf, B.: Measuring statistical dependence with Hilbert-Schmidt norms. In: Jain, S., Simon, H.U., Tomita, E. (eds.) ALT 2005. LNCS (LNAI), vol. 3734, pp. 63–77. Springer, Heidelberg (2005). https://doi.org/10.1007/11564089_7
45. Grömping, U.: Model-agnostic effects plots for interpreting machine learning models. Reports in Mathematics, Physics and Chemistry Report 1/2020 (2020)
46. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *J. Mach. Learn. Res.* **3**(Mar), 1157–1182 (2003)
47. Hall, P.: On the art and science of machine learning explanations. arXiv preprint [arXiv:1810.02909](https://arxiv.org/abs/1810.02909) (2018)
48. Hancox-Li, L.: Robustness in machine learning explanations: does it matter? In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, FAT* 2020, pp. 640–647. Association for Computing Machinery, New York (2020). <https://doi.org/10.1145/3351095.3372836>
49. Hand, D.J.: Classifier technology and the illusion of progress. *Stat. Sci.* **21**(1), 1–14 (2006). <https://doi.org/10.1214/088342306000000060>
50. Hastie, T., Tibshirani, R.: Generalized additive models. *Stat. Sci.* **1**(3), 297–310 (1986). <https://doi.org/10.1214/ss/1177013604>
51. He, Z., Yu, W.: Stable feature selection for biomarker discovery. *Comput. Biol. Chem.* **34**(4), 215–225 (2010). <https://doi.org/10.1016/j.compbiolchem.2010.07.002>
52. Holm, S.: A simple sequentially rejective multiple test procedure. *Scand. J. Stat.* **6**(2), 65–70 (1979)
53. Hooker, G.: Discovering additive structure in black box functions. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2004, pp. 575–580. Association for Computing Machinery, New York (2004). <https://doi.org/10.1145/1014052.1014122>
54. Hooker, G.: Generalized functional ANOVA diagnostics for high-dimensional functions of dependent variables. *J. Comput. Graph. Stat.* **16**(3), 709–732 (2007). <https://doi.org/10.1198/106186007X237892>
55. Hooker, G., Mentch, L.: Please stop permuting features: an explanation and alternatives. arXiv preprint [arXiv:1905.03151](https://arxiv.org/abs/1905.03151) (2019)
56. Janzing, D., Minorics, L., Blöbaum, P.: Feature relevance quantification in explainable AI: a causality problem. arXiv preprint [arXiv:1910.13413](https://arxiv.org/abs/1910.13413) (2019)
57. Kadir, T., Brady, M.: Saliency, scale and image description. *Int. J. Comput. Vis.* **45**(2), 83–105 (2001). <https://doi.org/10.1023/A:1012460413855>
58. Karimi, A.H., Schölkopf, B., Valera, I.: Algorithmic recourse: from counterfactual explanations to interventions. [arXiv:2002.06278](https://arxiv.org/abs/2002.06278) (2020)
59. Khamis, H.: Measures of association: how to choose? *J. Diagn. Med. Sonography* **24**(3), 155–162 (2008). <https://doi.org/10.1177/8756479308317006>
60. Kohavi, R., John, G.H.: Wrappers for feature subset selection. *Artif. Intell.* **97**(1–2), 273–324 (1997)
61. König, G., Freiesleben, T., Bischl, B., Casalicchio, G., Grosse-Wentrup, M.: Decomposition of global feature importance into direct and associative components (DEDACT). arXiv preprint [arXiv:2106.08086](https://arxiv.org/abs/2106.08086) (2021)
62. König, G., Freiesleben, T., Grosse-Wentrup, M.: A causal perspective on meaningful and robust algorithmic recourse. arXiv preprint [arXiv:2107.07853](https://arxiv.org/abs/2107.07853) (2021)
63. König, G., Molnar, C., Bischl, B., Grosse-Wentrup, M.: Relative feature importance. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 9318–9325. IEEE (2021). <https://doi.org/10.1109/ICPR48806.2021.9413090>

64. Krishnan, M.: Against interpretability: a critical examination of the interpretability problem in machine learning. *Philos. Technol.* **33**(3), 487–502 (2019). <https://doi.org/10.1007/s13347-019-00372-9>
65. Kuhle, S., et al.: Comparison of logistic regression with machine learning methods for the prediction of fetal growth abnormalities: a retrospective cohort study. *BMC Pregnancy Childbirth* **18**(1), 1–9 (2018). <https://doi.org/10.1186/s12884-018-1971-2>
66. König, G., Grosse-Wentrup, M.: A Causal Perspective on Challenges for AI in Precision Medicine (2019)
67. Lang, M., et al.: MLR3: a modern object-oriented machine learning framework in R. *J. Open Source Softw.* (2019). <https://doi.org/10.21105/joss.01903>
68. Laugel, T., Lesot, M.J., Marsala, C., Renard, X., Detyniecki, M.: The dangers of post-hoc interpretability: unjustified counterfactual explanations. In: Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, pp. 2801–2807. International Joint Conferences on Artificial Intelligence Organization (2019)
69. Laugel, T., Renard, X., Lesot, M.J., Marsala, C., Detyniecki, M.: Defining locality for surrogates in post-hoc interpretability. arXiv preprint [arXiv:1806.07498](https://arxiv.org/abs/1806.07498) (2018)
70. Lauritsen, S.M., et al.: Explainable artificial intelligence model to predict acute critical illness from electronic health records. *Nat. Commun.* **11**(1), 1–11 (2020). <https://doi.org/10.1038/s41467-020-17431-x>
71. Lessmann, S., Baesens, B., Seow, H.V., Thomas, L.C.: Benchmarking state-of-the-art classification algorithms for credit scoring: an update of research. *Eur. J. Oper. Res.* **247**(1), 124–136 (2015). <https://doi.org/10.1016/j.ejor.2015.05.030>
72. Liebetrau, A.: Measures of Association. No. Bd. 32; Bd. 1983 in 07, SAGE Publications (1983)
73. Lipton, Z.C.: The mythos of model interpretability. *Queue* **16**(3), 31–57 (2018). <https://doi.org/10.1145/3236386.3241340>
74. Lopez-Paz, D., Hennig, P., Schölkopf, B.: The randomized dependence coefficient. In: *Advances in Neural Information Processing Systems*, pp. 1–9 (2013). <https://doi.org/10.5555/2999611.2999612>
75. Lozano, A.C., Abe, N., Liu, Y., Rosset, S.: Grouped graphical granger modeling for gene expression regulatory networks discovery. *Bioinformatics* **25**(12), i110–i118 (2009). <https://doi.org/10.1093/bioinformatics/btp199>
76. Lundberg, S.M., et al.: From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.* **2**(1), 56–67 (2020). <https://doi.org/10.1038/s42256-019-0138-9>
77. Lundberg, S.M., Erion, G.G., Lee, S.I.: Consistent individualized feature attribution for tree ensembles. arXiv preprint [arXiv:1802.03888](https://arxiv.org/abs/1802.03888) (2018)
78. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: *NIPS*, vol. 30, pp. 4765–4774. Curran Associates, Inc. (2017). <https://doi.org/10.5555/3295222.3295230>
79. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: Statistical and machine learning forecasting methods: concerns and ways forward. *PloS One* **13**(3) (2018). <https://doi.org/10.1371/journal.pone.0194889>
80. Matejka, J., Fitzmaurice, G.: Same stats, different graphs: generating datasets with varied appearance and identical statistics through simulated annealing. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 1290–1294 (2017). <https://doi.org/10.1145/3025453.3025912>

81. Molnar, C., Casalicchio, G., Bischl, B.: IML: an R package for interpretable machine learning. *J. Open Source Softw.* **3**(26), 786 (2018). <https://doi.org/10.21105/joss.00786>
82. Molnar, C., Casalicchio, G., Bischl, B.: Quantifying model complexity via functional decomposition for better post-hoc interpretability. In: Cellier, P., Driessens, K. (eds.) *ECML PKDD 2019. CCIS*, vol. 1167, pp. 193–204. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-43823-4_17
83. Molnar, C., Freiesleben, T., König, G., Casalicchio, G., Wright, M.N., Bischl, B.: Relating the partial dependence plot and permutation feature importance to the data generating process. *arXiv preprint arXiv:2109.01433* (2021)
84. Molnar, C., König, G., Bischl, B., Casalicchio, G.: Model-agnostic feature importance and effects with dependent features—a conditional subgroup approach. *arXiv preprint arXiv:2006.04628* (2020)
85. Moosbauer, J., Herbinger, J., Casalicchio, G., Lindauer, M., Bischl, B.: Towards explaining hyperparameter optimization via partial dependence plots. In: 8th ICML Workshop on Automated Machine Learning (AutoML) (2020)
86. Mothilal, R.K., Sharma, A., Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. *CoRR abs/1905.07697* (2019). <http://arxiv.org/abs/1905.07697>
87. Oh, S.: Feature interaction in terms of prediction performance. *Appl. Sci.* **9**(23) (2019). <https://doi.org/10.3390/app9235191>
88. Pearl, J., Mackenzie, D.: *The Ladder of Causation. The Book of Why: The New Science of Cause and Effect*, pp. 23–52. Basic Books, New York (2018). <https://doi.org/10.1080/14697688.2019.1655928>
89. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011). <https://doi.org/10.5555/1953048.2078195>
90. Perneger, T.V.: What’s wrong with Bonferroni adjustments. *BMJ* **316**(7139), 1236–1238 (1998). <https://doi.org/10.1136/bmj.316.7139.1236>
91. Peters, J., Janzing, D., Scholkopf, B.: *Elements of Causal Inference - Foundations and Learning Algorithms*. The MIT Press (2017). <https://doi.org/10.5555/3202377>
92. Philipp, M., Rusch, T., Hornik, K., Strobl, C.: Measuring the stability of results from supervised statistical learning. *J. Comput. Graph. Stat.* **27**(4), 685–700 (2018). <https://doi.org/10.1080/10618600.2018.1473779>
93. Reshef, D.N., et al.: Detecting novel associations in large data sets. *Science* **334**(6062), 1518–1524 (2011). <https://doi.org/10.1126/science.1205438>
94. Ribeiro, M.T., Singh, S., Guestrin, C.: Why should I trust you?: explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144. ACM (2016). <https://doi.org/10.1145/2939672.2939778>
95. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.* **1**(5), 206–215 (2019). <https://doi.org/10.1038/s42256-019-0048-x>
96. Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., Zhong, C.: Interpretable machine learning: fundamental principles and 10 grand challenges. *arXiv preprint arXiv:2103.11251* (2021)
97. Saito, S., Chua, E., Capel, N., Hu, R.: Improving lime robustness with smarter locality sampling. *arXiv preprint arXiv:2006.12302* (2020)
98. Schallner, L., Rabold, J., Scholz, O., Schmid, U.: Effect of superpixel aggregation on explanations in lime—a case study with biological data. *arXiv preprint arXiv:1910.07856* (2019)

99. Schmid, M., Hothorn, T.: Boosting additive models using component-wise p-splines. *Comput. Stat. Data Anal.* **53**(2), 298–311 (2008). <https://doi.org/10.1016/j.csda.2008.09.009>
100. Scholbeck, C.A., Molnar, C., Heumann, C., Bischl, B., Casalicchio, G.: Sampling, intervention, prediction, aggregation: a generalized framework for model-agnostic interpretations. In: Cellier, P., Driessens, K. (eds.) *ECML PKDD 2019. CCIS*, vol. 1167, pp. 205–216. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-43823-4_18
101. Seedorff, N., Brown, G.: Totalvis: a principal components approach to visualizing total effects in black box models. *SN Comput. Sci.* **2**(3), 1–12 (2021). <https://doi.org/10.1007/s42979-021-00560-5>
102. Semenova, L., Rudin, C., Parr, R.: A study in Rashomon curves and volumes: a new perspective on generalization and model simplicity in machine learning. arXiv preprint [arXiv:1908.01755](https://arxiv.org/abs/1908.01755) (2021)
103. Shalev-Shwartz, S., Ben-David, S.: *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, Cambridge (2014)
104. Simon, R.: Resampling strategies for model assessment and selection. In: Dubitzky, W., Granzow, M., Berrar, D. (eds.) *Fundamentals of Data Mining in Genomics and Proteomics*, pp. 173–186. Springer, Cham (2007). https://doi.org/10.1007/978-0-387-47509-7_8
105. Stachl, C., et al.: Behavioral patterns in smartphone usage predict big five personality traits. *PsyArXiv* (2019). <https://doi.org/10.31234/osf.io/ks4vd>
106. Stachl, C., et al.: Predicting personality from patterns of behavior collected with smartphones. *Proc. Natl. Acad. Sci.* (2020). <https://doi.org/10.1073/pnas.1920484117>
107. Strobl, C., Boulesteix, A.L., Kneib, T., Augustin, T., Zeileis, A.: Conditional variable importance for random forests. *BMC Bioinform.* **9**(1), 307 (2008). <https://doi.org/10.1186/1471-2105-9-307>
108. Štrumbelj, E., Kononenko, I.: Explaining prediction models and individual predictions with feature contributions. *Knowl. Inf. Syst.* **41**(3), 647–665 (2013). <https://doi.org/10.1007/s10115-013-0679-x>
109. Sundararajan, M., Najmi, A.: The many Shapley values for model explanation. arXiv preprint [arXiv:1908.08474](https://arxiv.org/abs/1908.08474) (2019)
110. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: *International Conference on Machine Learning*, pp. 3319–3328. PMLR (2017)
111. Székely, G.J., Rizzo, M.L., Bakirov, N.K., et al.: Measuring and testing dependence by correlation of distances. *Ann. Stat.* **35**(6), 2769–2794 (2007). <https://doi.org/10.1214/009053607000000505>
112. Tibshirani, R.: Regression shrinkage and selection via the lasso. *J. Roy. Stat. Soc.: Ser. B (Methodol.)* **58**(1), 267–288 (1996). <https://doi.org/10.1111/j.1467-9868.2011.00771.x>
113. Tjostheim, D., Otneim, H., Støve, B.: Statistical dependence: beyond pearson’s p . arXiv preprint [arXiv:1809.10455](https://arxiv.org/abs/1809.10455) (2018)
114. Valentin, S., Harkotte, M., Popov, T.: Interpreting neural decoding models using grouped model reliance. *PLoS Comput. Biol.* **16**(1), e1007148 (2020). <https://doi.org/10.1371/journal.pcbi.1007148>
115. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: automated decisions and the GDPR. *Harv. JL Tech.* **31**, 841 (2017). <https://doi.org/10.2139/ssrn.3063289>

116. Walters-Williams, J., Li, Y.: Estimation of mutual information: a survey. In: Wen, P., Li, Y., Polkowski, L., Yao, Y., Tsumoto, S., Wang, G. (eds.) RSKT 2009. LNCS (LNAI), vol. 5589, pp. 389–396. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02962-2_49
117. Watson, D.S., Wright, M.N.: Testing conditional independence in supervised learning algorithms. arXiv preprint [arXiv:1901.09917](https://arxiv.org/abs/1901.09917) (2019)
118. Weichwald, S., Meyer, T., Özdenizci, O., Schölkopf, B., Ball, T., Grosse-Wentrup, M.: Causal interpretation rules for encoding and decoding models in neuroimaging. *Neuroimage* **110**, 48–59 (2015). <https://doi.org/10.1016/j.neuroimage.2015.01.036>
119. Williamson, B.D., Gilbert, P.B., Simon, N.R., Carone, M.: A unified approach for inference on algorithm-agnostic variable importance. [arXiv:2004.03683](https://arxiv.org/abs/2004.03683) (2020)
120. Wu, J., Roy, J., Stewart, W.F.: Prediction modeling using EHR data: challenges, strategies, and a comparison of machine learning approaches. *Med. Care* S106–S113 (2010). <https://doi.org/10.1097/MLR.0b013e3181de9e17>
121. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc.: Ser. B (Statistical Methodology)* **68**(1), 49–67 (2006). <https://doi.org/10.1111/j.1467-9868.2005.00532.x>
122. Zhang, X., Wang, Y., Li, Z.: Interpreting the black box of supervised learning models: visualizing the impacts of features on prediction. *Appl. Intell.* **51**(10), 7151–7165 (2021). <https://doi.org/10.1007/s10489-021-02255-z>
123. Zhao, Q., Hastie, T.: Causal interpretations of black-box models. *J. Bus. Econ. Stat.* 1–10 (2019). <https://doi.org/10.1080/07350015.2019.1624293>
124. Zhao, X., Lovreglio, R., Nilsson, D.: Modelling and interpreting pre-evacuation decision-making using machine learning. *Autom. Constr.* **113**, 103140 (2020). <https://doi.org/10.1016/j.autcon.2020.103140>
125. van der Zon, S.B., Duivesteyn, W., van Ipenburg, W., Veldsink, J., Pechenizkiy, M.: ICIE 1.0: a novel tool for interactive contextual interaction explanations. In: Alzate, C., et al. (eds.) MIDAS/PAP -2018. LNCS (LNAI), vol. 11054, pp. 81–94. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-13463-1_6

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Part III.

**Grouping Approaches in Interpretable
Machine Learning**

4. Grouped Feature Importance and Combined Features Effect Plot

In this article, we provide an overview of existing solutions and propose new solutions for the first limitation of global interpretation methods stated in Section 1.1, namely the *human incomprehensibility of high-dimensional output*. The suggested methods interpret feature groups instead of single features and thus reduce dimensionality and simplify the resulting output, which may enhance comprehensibility.

Contributing article: Au, Q., Herbinger, J., Stachl, C., Bischl, B., and Casalicchio, G. (2022). Grouped feature importance and combined features effect plot. *Data Mining and Knowledge Discovery* 36(4), 1401–1450. <https://doi.org/10.1007/s10618-022-00840-5>.

Author contributions: Julia Herbinger and Quay Au share the first authorship of this paper. Their overall equal contributions can be described as follows:

Quay Au and Giuseppe Casalicchio developed the project idea with support from Julia Herbinger and Bernd Bischl. Quay Au and Bernd Bischl developed the idea of the greedy forward search algorithm to select the most important feature groups. The idea of calculating Shapley feature importance for groups of features and their decomposition into single feature importance values was developed and proven by Quay Au, Julia Herbinger, and Giuseppe Casalicchio with equal contributions. The idea and the algorithm of the combined features effect plot were developed by Quay Au, Julia Herbinger, and Giuseppe Casalicchio with equal contributions. All algorithms were implemented by Quay Au and revised and improved by Julia Herbinger. Julia Herbinger designed and conducted the experiments with continuous support from Quay Au and Giuseppe Casalicchio. Quay Au conducted the real-world analysis with support from Clemens Stachl. The manuscript was drafted jointly by Quay Au and Julia Herbinger with overall equal contributions. All authors contributed to revisions of the paper. Giuseppe Casalicchio, Clemens Stachl, and Bernd Bischl gave valuable input throughout the project and suggested several notable modifications.

Data Mining and Knowledge Discovery (2022) 36:1401–1450
<https://doi.org/10.1007/s10618-022-00840-5>



Grouped feature importance and combined features effect plot

Quay Au¹ · Julia Herbinger¹ · Clemens Stachl² · Bernd Bischl¹ · Giuseppe Casalicchio¹

Received: 23 April 2021 / Accepted: 18 April 2022 / Published online: 18 June 2022
© The Author(s) 2022

Abstract

Interpretable machine learning has become a very active area of research due to the rising popularity of machine learning algorithms and their inherently challenging interpretability. Most work in this area has been focused on the interpretation of single features in a model. However, for researchers and practitioners, it is often equally important to quantify the importance or visualize the effect of feature groups. To address this research gap, we provide a comprehensive overview of how existing model-agnostic techniques can be defined for feature groups to assess the grouped feature importance, focusing on permutation-based, refitting, and Shapley-based methods. We also introduce an importance-based sequential procedure that identifies a stable and well-performing combination of features in the grouped feature space. Furthermore, we introduce the combined features effect plot, which is a technique to visualize the effect of a group of features based on a sparse, interpretable linear com-

Responsible editor: Martin Atzmueller, Johannes Fürnkranz, Tomáš Kliegr and Ute Schmid.

Quay Au and Julia Herbinger have contributed equally to this work.

✉ Julia Herbinger
julia.herbinger@stat.uni-muenchen.de

Quay Au
quayau@gmail.com

Clemens Stachl
clemens.stachl@unisg.ch

Bernd Bischl
bernd.bischl@stat.uni-muenchen.de

Giuseppe Casalicchio
giuseppe.casalicchio@stat.uni-muenchen.de

¹ Department of Statistics, Ludwig-Maximilians-University Munich, 80539 Munich, Germany

² Institute of Behavioral Science and Technology, University of St. Gallen, 9000 St. Gallen, Switzerland

bination of features. We used simulation studies and real data examples to analyze, compare, and discuss these methods.

Keywords Grouped feature importance · Combined features effects · Dimension reduction · Interpretable machine learning

1 Introduction

Machine learning (ML) algorithms are nowadays used in many diverse fields e.g. in medicine (Shipp et al. 2002), criminology (Berk et al. 2009), and increasingly in the social sciences (Stachl et al. 2020b; Yarkoni and Westfall 2017). Interpretable models are paramount in many high-stakes settings, such as medical and juridical applications (Lipton 2018). However, well-performing ML models often bear a lack of interpretability. In the context of interpretable ML (IML) research, several model-agnostic methods to produce explanations for single features have been developed (Molnar 2019). Examples include the permutation feature importance (PFI; Fisher et al. 2019), leave-one-covariate out (LOCO) importance (Lei et al. 2018), SHAP values (Lundberg and Lee 2017), or partial dependence plots (PDP; Friedman 2001).

In many applications, it can be more informative to produce explanations for the importance or effect of a group of features (which we refer to as grouped interpretations) rather than for single features. It is important to note that the meaning of grouped interpretations, in general, differs from single feature interpretations, and resulting interpretations are usually not directly comparable (e.g., as Gregorutti et al. (2015) shows for the permutation feature importance). Hence, our aim is not to challenge single feature interpretations as both single and grouped feature interpretation methods measure different things and are useful on their own.

Grouped interpretations might be especially interesting for high-dimensional settings with hundreds or thousands of features. In particular, when analyzing the influence of these features visually (e.g., by plotting the marginal effect of a feature on the target) on a single feature level, this might result in an information overload which might not provide a comprehensive understanding of the learned effects (Molnar et al. 2020b). Furthermore, the runtime of some interpretation methods—such as Shapley values—does not scale linearly in the number of features. Hence, calculating them on a single feature level might not be computationally feasible for high-dimensional settings, making grouped computations a feasible remedy (Lundberg and Lee 2017; Covert et al. 2020; Molnar et al. 2020b).

From a use case perspective, the concept of grouped interpretations is particularly useful when the feature grouping is available *a priori* based on the application context. In that sense, features that either belong to the same semantic area (e.g., behaviors in psychology or biomarkers in medicine) or are generated by the same mechanism or device (e.g., fMRI, EEG, smartphones) can be grouped together to assess their joint effect or importance. For example, in our application in Sect. 7, we use a real-world use case from psychology that studies how the human behavior on smartphone app usage is associated to different personality traits (Stachl et al. 2020a). Features were extracted from longitudinal data collected from smartphones of 624 participants,



Fig. 1 A possible process from group definition to grouped interpretations. First, the feature groups must be defined. A model is then fitted, typically on the feature space where the information of the pre-defined grouping might be used (e.g., if the fitting process is combined with a feature selection procedure) or ignored. When the best model is found, model-agnostic grouped interpretation methods are applied on the previously defined feature groups. A commonly used approach is to first obtain an overview of which groups are most important for achieving a good model performance (grouped feature importance) to subsequently analyze how the most important feature groups influence the model's prediction (grouped feature effect) (Color figure online)

and can be grouped into different behavioral classes (i.e., communication and social activity, app-usage, music consumption, overall phone activity, mobility). Another example is applications with sensor data (Chakraborty and Pal 2008), where multiple features measured by a single sensor naturally belong together, and hence grouped interpretations on sensor-level might be more informative.

There are also situations where the interpretation of single features might be misleading and where grouped interpretations can provide a remedy. Examples include datasets with time-lagged or categorical features (e.g., dummy or one-hot encoded categories) and the presence of feature interactions (Gregorutti et al. 2015). A concrete example for dummy encoded categorical features is shown in Appendix A.

Even in situations where feature groups are not naturally given in advance, it still might be beneficial to define groups in a data-driven manner and apply interpretation methods on groups of features (for examples, see Sect. 1.2).

Hence, compared to single feature interpretation methods, the grouping structure must be defined beforehand. A possible process—from group membership definition to modeling up to post-hoc interpretations—is illustrated in Fig. 1. Since defining the underlying group structure is a relevant step in this process, we discuss some applied techniques on how to find groups of features in Sect. 1.2. However, in this paper, we focus on the interpretation component once the groups are known (the green part in Fig. 1).

Although the grouped feature perspective is relevant in many applications, most IML research has focused on methods that attempt to provide explanations on a single-feature level. Model-agnostic methods for feature groups are rare and not well-studied.

1.1 Real data use cases with grouped features

In the following we summarize further exemplary predictive tasks with pre-specified feature groupings. These tasks will also be used in Sect. 3.4 for further empirical analysis. For more details on features and associated groups see Table 1.

Heat value of fossil fuels In this small scale regression task ($n = 129$), the objective is to predict the heat value of fossil fuels from spectral data (Fuchs et al. 2015). In addition to one scalar feature (humidity), the dataset contains two groups of curve data, the first from the ultraviolet-visible spectrum (UVVIS) and the second from the near infrared spectrum (NIR).

Table 1 Real world datasets with grouped features and their pre-specified group memberships

Dataset	Single features	Group membership	Description
<i>Birthweight</i>	age1, age2, age3	Age	Mother's age represented by 3 orthogonal polynomials
	lwt1, lwt2, lwt3	lwt	Mother's weight represented by 3 orthogonal polynomials
	White, black	Race	Mother's race (indicator functions)
	Smoke	Smoke	Smoking status (indicator function)
	ptl1, ptl2m	ptl	One, or two or more previous premature labors
	ht	ht	History of hypertension (indicator function)
	ui	ui	Presence of uterine irritability (indicator function)
	ftv1, ftv2, ftv3m	ftv	One, two, or three or more physician visits during first trimester
<i>Colon</i>	x1, ..., x5	Gene1	Gene expression data for gene 1
	⋮	⋮	⋮
	x96, ..., x100	Gene20	Gene expression data for gene 20
<i>Fuelsubset</i>	H20	H20	Humidity in percent
	UVVIS1, ..., UVVIS134	UVVIS	Data from the ultraviolet-visible spectrum (134 wavelength points)
	NIR1, ..., NIR231	NIR	Data from the near infrared spectrum (231 wavelength points)

Birthweight The *birthweight* dataset has data on 189 births at the Baystate Medical Centre in Massachusetts during 1986 (Venables and Ripley 2002). The objective is to predict the birth weight in kilograms from a set of 16 features, some of which are grouped (e.g., dummy encoded categorical features).

Colon cancer The *colon* dataset contains gene expression data of 20 genes (5 basis B-Splines each) for 62 samples from microarray experiments of colon tissue (Alon et al. 1999). The task is to predict cancerous tissue from the resulting 100 predictors.

1.2 Grouping procedures

Following the definitions of He and Yu (2010), we provide a brief overview of different procedures to define feature groups in a knowledge-driven and data-driven manner. In data-driven grouping, an algorithmic approach such as clustering or density estimation is used to define groups of features. Knowledge-driven grouping, on the other hand, uses domain knowledge to define the grouping structure of features. Throughout our

paper, we mainly assume a user-defined grouping structure. However, all methods introduced in this paper should also be compatible with an appropriate data-driven method if the defined groups have a meaningful interpretation.

Data-driven grouping

One method to group features in a data-driven manner is to use clustering approaches such as hierarchical clustering (Park et al. 2006; Toloși and Lengauer 2011; Rapaport et al. 2008) or fuzzy clustering (Jaeger et al. 2003). These approaches often work well in highly correlated feature spaces, such as in genomics or medicine, where correlated features are grouped together so that no relevant information is discarded (Toloși and Lengauer 2011). For instance, Jaeger et al. (2003) tackles a feature selection problem for a high-dimensional and intercorrelated feature space when working with microarray data. To simultaneously select informative and distinct genes, they first apply fuzzy clustering to obtain groups of similar genes from microarray data. Next, the informative representatives of each group are selected based on a suitable test statistic. The disadvantage of data-driven grouping is that groups depend only on the statistical similarity between features, which might not coincide with domain-specific interpretations (Chakraborty and Pal 2008).

Knowledge-driven grouping

Knowledge-driven group formation has the advantage that the dimensionality reduction might lead to better interpretability than the data-driven path. Gregorutti et al. (2015) apply a knowledge-driven approach in the context of multiple functional data analysis, where they then select groups for subsequent modeling based on their group importance values. Chakraborty and Pal (2008) also select groups of features, where data from one sensor (e.g., to capture satellite images in different spectral bands) represents a group. Hence, features are grouped based on their topical character (e.g., measurement device) rather than their shared statistical properties. Another use case of knowledge-driven grouping is described in Lozano et al. (2009), who group time-lagged features of the same time series for gene expression data. They use the given grouping structure in a group feature selection procedure and apply group LASSO as well as a boosting method.

1.3 Related work

A well-known model that handles groups of features is the *group LASSO* (Yuan and Lin 2006), which extends the LASSO (Tibshirani 1996) for feature selection based on groups. Moreover, other extensions—e.g., to obtain sparse groups of features (Friedman et al. 2010), to support classification tasks (Meier et al. 2008) or non-linear effects (Gregorova et al. 2018)—also exist. However, group LASSO is a modeling technique that focuses on selecting groups in the feature space rather than quantifying their importance.

A large body of research already exists regarding the importance of individual features (see, e.g., Fisher et al. 2019; Hooker and Mentch 2019; Scholbeck et al. 2020). Hooker and Mentch (2019) distinguish between two loss-based feature importance approaches, namely permutation methods and refitting methods. Permutation meth-

ods measure the increase in expected loss (or error) after permuting a feature while the model remains untouched. Refitting methods measure the increase in expected loss after leaving out the feature of interest completely and hence require refitting the model (Lei et al. 2018). Since the model remains untouched in the former approach, interpretations refer to a specific fitted model, while interpretations for refitting methods refer to the underlying ML algorithm. Gregorutti et al. (2015) introduced a model-specific, grouped PFI score for random forests and applied this approach to functional data analysis. Valentin et al. (2020) introduced a model-agnostic grouped version of the model reliance score (Fisher et al. 2019). However, they focus more on the application and omit a detailed theoretical foundation. Recently, a general refitting framework to measure the importance of (groups of) features was introduced by Williamson et al. (2020). In their approach, the feature importance measurement is detached from the model level and defined by an algorithm-agnostic version to measure the intrinsic importance of features. The importance score is defined as the difference between the performance of the full model and the performance based on all features *except* the group of interest.

Permutation methods can be computed much faster than refitting methods. However, the PFI, for example, has issues when features are correlated and interact in the model due to extrapolation in regions without any or just a few observations (Hooker and Mentch 2019). Hence, interpretations in these regions might be misleading. To avoid this problem, alternatives based on conditional distributions or refitting have been suggested (e.g., Strobl et al. 2008; Nicodemus et al. 2010; Hooker and Mentch 2019; Watson and Wright 2019; Molnar et al. 2020a). Although the so-called conditional PFI provides a solution to this problem, its interpretation is different and “must be interpreted as the additional, unique contribution of a feature given all remaining features we condition on were known” (Molnar et al. 2020a). This property complicates the comparison with non-conditional interpretation methods. Therefore, we do not consider any conditional variants in this paper.

A third class of importance measures is based on Shapley values (Shapley 1953), a theoretical concept of game theory. The SHAP (Lundberg and Lee 2017) approach quantifies the contribution of each feature to the predicted outcome and is a permutation-based method. It has the advantage that contributions of interactions are distributed fairly between features. Besides being computationally more expensive, SHAP itself is based on the model’s predicted outcome rather than the model’s performance (e.g., measured by the model’s expected loss). Casalicchio et al. (2019) extended the concept of Shapley values to fairly distribute the model’s performance among features and called it Shapley Feature IMPortance (SFIMP). A similar approach called SAGE has also been proposed by Covert et al. (2020), who showed the benefits of the method on various simulation studies. One approach that uses Shapley values to explain grouped features was introduced by de Mijolla et al. (2020). However, instead of directly computing Shapley importance on the original feature space, they first apply a semantically-meaningful latent representation (e.g. by projecting the original feature space into a lower dimensional latent variable space using disentangled representations) and compute the Shapley importance on the resulting latent variables. Williamson and Feng (2020) mention that their feature importance method based on Shapley values can also be extended to groups of features. Additionally,

Amoukou et al. (2021) investigated grouping approaches for Shapley values in the case of encoded categorical features and subset selection of important features for tree-based methods. The calculation of Shapley values on groups of features based on performance values has only been applied with regard to feature subset selection methods and not for interpretation purposes (Cohen et al. 2005; Tripathi et al. 2020).¹

After identifying which groups of features are important, the user is often interested in how they (especially the important groups) influence the model's prediction. Several techniques to visualize single-feature effects exist. These include partial dependence plots (PDP) (Friedman 2001), individual conditional expectation (ICE) curves (Goldstein et al. 2013), SHAP dependence plots (Lundberg et al. 2018), and accumulated local effects (ALE) plots (Apley and Zhu 2019). However, in the case of high-dimensional feature spaces, it is often not feasible to compute, visualize, and interpret single-feature plots for all (important) features. If features are grouped, visualization techniques become computationally more complex, and it may become even harder to visualize the results in an easily interpretable way. In the case of low-dimensional feature spaces, this might still be feasible, for example by using two-feature PDPs or ALE plots. Recently, effect plots that visualize the combined effect of multiple features have been introduced by Seedorff and Brown (2021) and Brenning (2021). They use principal component analysis (PCA) to reduce the dimension of the feature space and calculate marginal effect curves for the principal components. However, the employed dimension reduction method does not include information about the target variable and lacks sparsity (and hence, interpretability).

1.4 Contribution

Our contributions can be summarized as follows: We extend the permutation-based and refitting-based grouped feature importance methods introduced by Valentin et al. (2020) and Williamson et al. (2020) by comparing these methods to not only the full model (i.e., taking into account all features), but also to a null model (i.e., ignoring all features). Hence, we can quantify to what extent a group itself contributes to the prediction of a model without the presence of other groups. Furthermore, we introduce Shapley importance for feature groups and describe how these scores can be decomposed into single-feature importance scores of the respective groups. Our main contributions are: (1) We define a new algorithm to sequentially add groups of features depending on their importance, thereby enabling identification of well-performing combinations of groups. (2) We compare all grouped feature importance methods with respect to the main challenges that arise when applying these methods by creating small simulation examples. Subsequently, we provide recommendations for using and interpreting the respective methods correctly. (3) We introduce a model-agnostic method to visualize the joint effect of a group of features. To that end, we use a suitable dimension reduction technique and the conceptual idea of PDPs to calculate and plot the mean prediction of a sparse group of features with regard to their linear

¹ Feature subset selection methods usually aim to find sparse, well-performing feature combinations. Hence, the intended purpose of employing these methods is not to produce interpretability, but rather to generate a sufficient performance with fewer features.

combination. This novel method finally enables the user to visualize effects for groups of features. Finally, we showcase the usefulness of all these methods in real data examples.

The structure of this paper is as follows: First, we provide some general notation and definitions in Sect. 2. We formally define the grouped feature importance methods and introduce the sequential grouped feature importance procedure in Sects. 3 and 4, respectively. We compare these methods for different scenarios in Sect. 5. In Sect. 6, we introduce the combined features effect plot (CFEP) to visualize the effects of feature groups based on a supervised dimension reduction technique. Moreover, we also show the suitability of this technique compared to its unsupervised counterpart in a simulation study. Finally, in Sect. 7, all methods are applied to a real data example before summarizing and offering an outlook for future research in Sect. 8.

2 Background and notation

Analogous to Casalicchio et al. (2019), we use the term *feature importance* to refer to the influence of features on a model's predictive performance, which we measure by the expected loss when we perturb these features in a permutation approach or remove these features in a refitting approach.

2.1 General notation

Consider a p -dimensional feature space $\mathcal{X} = (\mathcal{X}_1 \times \dots \times \mathcal{X}_p)$ and a one-dimensional target space \mathcal{Y} . The corresponding random variables that are generated from these spaces are denoted by $X = (X_1, \dots, X_p)$ and Y . We denote a ML prediction function that maps the p -dimensional feature space to a one-dimensional target space by $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}$ for regression tasks.² ML algorithms try to learn this functional relationship using $n \in \mathbb{N}$ i.i.d. observations drawn from the joint space $\mathcal{X} \times \mathcal{Y}$ with unknown probability distribution \mathcal{P} . The resulting dataset is denoted by $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$, where the vector $\mathbf{x}^{(i)} = (x_1^{(i)}, \dots, x_p^{(i)})^\top \in \mathcal{X}$ is the i -th observation associated with the target variable $y^{(i)} \in \mathcal{Y}$. The j -th feature is denoted by $\mathbf{x}_j = (x_j^{(1)}, \dots, x_j^{(n)})^\top$, for $j = 1, \dots, p$. The dataset \mathcal{D} can also be written in matrix form:

$$\begin{pmatrix} x_1^{(1)} & \dots & x_p^{(1)} & y^{(1)} \\ \vdots & \ddots & \vdots & \vdots \\ x_1^{(n)} & \dots & x_p^{(n)} & y^{(n)} \end{pmatrix} = (\mathbf{X}, \mathbf{Y}), \text{ with } \mathbf{X} = \begin{pmatrix} x_1^{(1)} & \dots & x_p^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(n)} & \dots & x_p^{(n)} \end{pmatrix}, \mathbf{Y} = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{pmatrix}. \quad (1)$$

The general error measure $\rho(\hat{f}, \mathcal{P}) = \mathbb{E}(L(\hat{f}(X), Y))$ of a learned model \hat{f} is measured by a loss function L on test data drawn independently from \mathcal{P} and can be

² The target space is defined by \mathbb{R}^g in the case of scoring classifiers with g classes.

estimated using unseen test data $\mathcal{D}_{\text{test}}$ by

$$\hat{\rho}(\hat{f}, \mathcal{D}_{\text{test}}) = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{(\mathbf{x}, y) \in \mathcal{D}_{\text{test}}} L(\hat{f}(\mathbf{x}), y). \quad (2)$$

The application of an ML algorithm (or *learner*) \mathcal{I} to a given dataset \mathcal{D} results in a fitted model $\mathcal{I}(\mathcal{D}) = \hat{f}_{\mathcal{D}}$. The *expected generalization error* of a learner \mathcal{I} takes into account the variability introduced by sampling different datasets \mathcal{D} of equal size n from \mathcal{P} and is defined by

$$GE(\mathcal{I}, \mathcal{P}, n) = \mathbb{E}_{|\mathcal{D}|=n}(\rho(\mathcal{I}(\mathcal{D}), \mathcal{P})). \quad (3)$$

In practice, resampling techniques such as cross-validation or bootstrapping on the available dataset \mathcal{D} are used to estimate Eq. (3). Resampling techniques usually split the dataset \mathcal{D} into $k \in \mathbb{N}$ training datasets $\mathcal{D}_{\text{train}}^i$, $i = 1, \dots, k$, of roughly the same size $n_{\text{train}} < n$. Eq. (3) can be estimated by

$$\widehat{GE}(\mathcal{I}, \mathcal{D}, n_{\text{train}}) = \frac{1}{k} \sum_{i=1}^k \hat{\rho}(\hat{f}_{\mathcal{D}_{\text{train}}^i}, \mathcal{D}_{\text{test}}^i). \quad (4)$$

In the following, we often associate the set of numbers $\{1, \dots, p\}$ in a one-to-one manner with the features $\mathbf{x}_1, \dots, \mathbf{x}_p$ by referring a number $j \in \{1, \dots, p\}$ as feature x_j . We call $G \subset \{1, \dots, p\}$ a *group of features*.

2.2 Permutation feature importance (PFI)

Fisher et al. (2019) proposed a model-agnostic version of the PFI measure used in random forests (Breiman 2001). The PFI score of the j -th feature of a fitted model \hat{f} is defined as the increase in expected loss after permuting feature X_j :

$$\text{PFI}_j(\hat{f}) = \mathbb{E}(L(\hat{f}(X_{[j]}), Y)) - \mathbb{E}(L(\hat{f}(X), Y)). \quad (5)$$

Here, $X_{[j]} = (X_1, \dots, X_{j-1}, \tilde{X}_j, X_{j+1}, \dots, X_p)$ is the p -dimensional random variable vector of features, where \tilde{X}_j is an independent replication of X_j following the same distribution. The idea behind this method is to break the association between the j -th feature and the target variable by permuting its feature values. If a feature is not useful for predicting an outcome, changing its values by permutation will not increase the expected loss.³ For an accurate estimation of Eq. (5), we would need to calculate all possible permutation vectors over the index set $\{1, \dots, n\}$ (see Casalicchio et al. (2019) for an in-depth discussion on this topic). However, Eq. (5) can be approximated on a dataset \mathcal{D} with n observations by Monte Carlo integration using m

³ We consider the case of loss functions that are to be minimized. Hence, the larger PFI_j , the more substantial the increase in expected loss and the more important the j -th feature.

random permutations:

$$\widehat{\text{PFI}}_j(\hat{f}, \mathcal{D}) = \frac{1}{nm} \sum_{i=1}^n \sum_{k=1}^m \left(L\left(\hat{f}(x_1^{(i)}, \dots, x_j^{\tau_k^{(i)}}, \dots, x_p^{(i)}), y^{(i)}\right) - L(\hat{f}(\mathbf{x}^{(i)}, y^{(i)})) \right), \quad (6)$$

where τ_k is a random permutation vector of the index set $\{1, \dots, n\}$ for $k = 1, \dots, m$ permutations.⁴

Equation (6) could also be embedded into a resampling technique, where the permutation is always applied on the held-out test set of each resampling iteration (Fisher et al. 2019). However, this leads to refits and is computationally more expensive. The resulting resampling-based PFI of a learner \mathcal{I} is estimated by

$$\widehat{\text{PFI}}_j^{\text{res}}(\mathcal{I}, \mathcal{D}, n_{\text{train}}) = \frac{1}{k} \sum_{i=1}^k \widehat{\text{PFI}}_j(\hat{f}_{\mathcal{D}_{\text{train}}^i}, \mathcal{D}_{\text{test}}^i), \quad (7)$$

where the permutation strategy is applied on the test sets $\mathcal{D}_{\text{test}}^i$.

3 Feature importance for groups

In our first minor contribution, we provide a general notation and formal definitions for grouped permutation and refitting methods and explain them by answering the following questions:

- To what extent does a group of features contribute to the model's performance in the presence of other groups?
- To what extent does a group itself increase the expected loss if it is added to a null model like the mean prediction of the target for refitting methods?
- How can we fairly distribute the expected loss among all groups and all features within a group?

The definitions of all grouped feature importance scores are based on loss functions. They are defined in such a way that important groups will yield positive grouped feature importance scores. The question of how to interpret the differing results of these methods is addressed in Sect. 5.

3.1 Permutation methods

Here, we extend the existing definition of PFI to groups of features and introduce the GPFI (Grouped Permutation Feature Importance) and GOPFI (Group Only Permutation Feature Importance) scores. For ease of notation, we will only define these scores for a fitted model \hat{f} (see Eq. 5).

⁴ An example for $n = 3$ would be $\tau_1 = (1, 3, 2)^T$ with $\tau_1^{(i)}$ being the i -th entry of that vector.

3.1.1 Grouped permutation feature importance (GPFI)

For the definition of GPFI—which is based on the definitions of Gregorutti et al. (2015) and Valentin et al. (2020)—let $G \subset \{1, \dots, p\}$ be a group of features. Let $\tilde{X}_G = (\tilde{X}_j)_{j \in G}$ be a $|G|$ -dimensional random vector of features, which is an independent replication of $X_G = (X_j)_{j \in G}$ following the same joint distribution. This random vector is independent of both the target variable and the random vector of the remaining features, which we define by $X_{-G} := (X_j)_{j \in \{1, \dots, p\} \setminus G}$. With slight abuse of notation to index the feature groups included in G , we define the grouped permutation feature importance of G as

$$\text{GPFI}_G = \mathbb{E}(L(\hat{f}(\tilde{X}_G, X_{-G}), Y)) - \mathbb{E}(L(\hat{f}(X), Y)). \quad (8)$$

Equation (8) extends Eq. (5) to groups of features so that the interpretation of GPFI scores always refers to the importance when the feature values of the group defined by G are permuted jointly (i.e., without destroying the dependencies of the features within the group). Similar to Eq. (7), the grouped permutation feature importance can be estimated by Monte Carlo integration:

$$\widehat{\text{GPFI}}_G = \frac{1}{nm} \sum_{i=1}^n \sum_{k=1}^m \left(L(\hat{f}(\mathbf{x}_G^{(\tau_k^{(i)})}, \mathbf{x}_{-G}^{(i)}), y^{(i)}) - L(\hat{f}(\mathbf{x}^{(i)}), y^{(i)}) \right). \quad (9)$$

The GPFI measures the contribution of one group to the model's performance if all other groups are present in the model (see (a) from Sect. 3).

3.1.2 Group only permutation feature importance (GOPFI)

To evaluate the extent to which a group itself contributes to a model's performance (see (b) from Sect. 3), one can also use a slightly different measure. As an alternative to Eq. 9, we can compare the expected loss after permuting all features jointly with the expected loss after permuting all features except the considered group. We define this GOPFI for a group $G \subset \{1, \dots, p\}$ as

$$\text{GOPFI}_G = \mathbb{E}(L(\hat{f}(\tilde{X}), Y)) - \mathbb{E}(L(\hat{f}(X_G, \tilde{X}_{-G}), Y)), \quad (10)$$

which can be approximated by

$$\widehat{\text{GOPFI}}_G = \frac{1}{nm} \sum_{j=1}^n \sum_{k=1}^m \left(L(\hat{f}(\mathbf{x}^{(\tau_k^{(j)})}, y^{(j)})) - L(\hat{f}(\mathbf{x}_G^{(j)}, \mathbf{x}_{-G}^{(\tau_k^{(j)})}), y^{(j)}) \right). \quad (11)$$

While the relevance of GOPFI as an importance measure might be limited, it is technically useful for the grouped Shapley importance (see Eq. 14).

3.2 Refitting methods

Here, we introduce two refitting-based methods for groups of features. The first definition is similar to the one introduced in Williamson et al. (2020).

3.2.1 Leave-one-group-out importance (LOGO)

For a subset $G \subset \{1, \dots, p\}$, we define the reduced dataset $\tilde{\mathcal{D}} := \{(\mathbf{x}_{-G}^{(i)}, y^{(i)})\}_{i=1}^n$. Given a learner \mathcal{I} , which generates models $\mathcal{I}(\mathcal{D}) = \hat{f}_{\mathcal{D}}$ and $\mathcal{I}(\tilde{\mathcal{D}}) = \hat{f}_{\tilde{\mathcal{D}}}$, we define the Leave-One-Group-Out Importance (LOGO) as

$$\text{LOGO}(G) = \mathbb{E}(L(\hat{f}_{\tilde{\mathcal{D}}}(X_{-G}), Y)) - \mathbb{E}(L(\hat{f}_{\mathcal{D}}(X), Y)). \quad (12)$$

The LOGO can be estimated by using a learner \mathcal{I} on $\tilde{\mathcal{D}}$ and should be embedded in a resampling technique:

$$\begin{aligned} \widehat{\text{LOGO}}(G) &= \widehat{\text{GE}}(\mathcal{I}, \tilde{\mathcal{D}}, n_{\text{train}}) - \widehat{\text{GE}}(\mathcal{I}, \mathcal{D}, n_{\text{train}}) \\ &= \frac{1}{k} \sum_{i=1}^k \hat{\rho}(\hat{f}_{\tilde{\mathcal{D}}_{\text{train}}^i}, \tilde{\mathcal{D}}_{\text{test}}^i) - \frac{1}{k} \sum_{i=1}^k \hat{\rho}(\hat{f}_{\mathcal{D}_{\text{train}}^i}, \mathcal{D}_{\text{test}}^i). \end{aligned}$$

Consequently, we compare the increase in expected loss compared to the full model's expected loss when leaving out a group of features and performing a refit (see (a) from Sect. 3).

While GPFI can be calculated with a resampling-based strategy by using refits to receive the algorithm-based instead of model-based GPFI, the meaning still varies from LOGO. For the algorithm-based GPFI, we calculate for each fitted model the importance score by permuting the regarded group and predicting with the same model. Then we average over all models from our resampling strategy and receive an importance score, which tells us how important a group of features is for some learner \mathcal{I} when we break the association between this group and all other groups and the target. LOGO, on the other hand, leaves the group out and then performs the refit to calculate the importance of the group, and hence, it addresses the question: Can we remove this group from our dataset without reducing our model's performance? This is not answered by permutation-based methods.

3.2.2 Leave-one-group-in importance (LOGI)

While it may be too limiting to estimate the performance of a model based on one feature only, it can be informative to determine the extent to which a group of features (e.g., all measurements from a specific medical device) can reduce the expected loss in contrast to a null model (see (b) from Sect. 3). The Leave-One-Group-In (LOGI) method could be particularly helpful in settings where information on additional groups of measures will induce significant costs (e.g., adding functional imaging

data for a diagnosis) and/or limited resources are available (e.g., in order to be cost-covering, only one group of measures can be acquired). The LOGI method can also be useful for theory development in the natural and social sciences (e.g., which group of behaviors is most predictive by itself).

Let $\mathcal{I}_{\text{null}}$ be a null algorithm, which results in a null model \hat{f}_{null} that only guesses the mean (or majority class for classification) of the target variable for any dataset. We additionally define a learner \mathcal{I} , which generates a model $\mathcal{I}(\hat{\mathcal{D}}) = \hat{f}_{\hat{\mathcal{D}}}$ for a dataset $\hat{\mathcal{D}} := \{(\mathbf{x}_G^{(i)}, y^{(i)})\}_{i=1}^n$, which only contains features defined by $G \subset \{1, \dots, p\}$. We define the LOGI of a group G as

$$\text{LOGI}(G) = \mathbb{E}(L(\hat{f}_{\text{null}}, Y)) - \mathbb{E}(L(\hat{f}_{\hat{\mathcal{D}}}(X_G), Y)). \tag{13}$$

The LOGI can be estimated by using a learner \mathcal{I} on $\hat{\mathcal{D}} = \{(\mathbf{x}_G^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^n$ and should be embedded in a resampling technique:

$$\begin{aligned} \widehat{\text{LOGI}}(G) &= \widehat{GE}(\mathcal{I}_{\text{null}}, \mathcal{D}, n_{\text{train}}) - \widehat{GE}(\mathcal{I}, \hat{\mathcal{D}}, n_{\text{train}}) \\ &= \frac{1}{k} \sum_{i=1}^k \hat{\rho}(\hat{f}_{\text{null}}, \mathcal{D}_{\text{test}}^i) - \frac{1}{k} \sum_{i=1}^k \hat{\rho}(\hat{f}_{\hat{\mathcal{D}}_{\text{train}}^i}, \hat{\mathcal{D}}_{\text{test}}^i). \end{aligned}$$

3.3 Grouped Shapley importance (GSI)

The importance measures defined above either exclude (or permute) individual groups of features from the total set of features or consider only the importance of groups by omitting (or permuting) all other features. The grouped importance scores are usually not affected if interactions within the groups are present. However, they can be affected if features from different groups interact, since permuting a group of features jointly destroys any interactions with other features outside the considered group. Therefore, we define the grouped Shapley importance (GSI) based on Shapley values (Shapley 1953). GSI scores account for feature interactions, as they measure the average contribution of a given group to all possible combinations of groups and fairly distribute the importance value caused by interactions among all groups (see (c) from Sect. 3).

We assume a set of distinct groups $\mathcal{G} = \{G_1, \dots, G_l\}$, with $G_i \subset \{1, \dots, p\}$, for $i = 1, \dots, l$. In our grouped feature context, the value function $v : \mathcal{P}(\mathcal{G}) \rightarrow \mathbb{R}$ assigns a ‘‘payout’’ to each possible group or combination of groups included in \mathcal{G} . With slight abuse of notation, we define the value function for a subset $S \subset \mathcal{G}$ as

$$v(S) := v(\cup_{G_i \in S} G_i).$$

We define the value function for a group $G \in \mathcal{G}$ calculated by a refitting or a permutation method by

$$v_{\text{refit}}(G) = \text{LOGI}(G) \quad \text{or} \quad v_{\text{perm}}(G) = \text{GOPFI}(G), \tag{14}$$

respectively. The marginal contribution of a group $G \in \mathcal{G}$, with $S \subset \mathcal{G}$ is

$$\Delta_G(S) = v(S \cup G) - v(S).$$

The GSI of the feature group G is then defined as

$$\phi(G) = \sum_{S \subset \mathcal{G} \setminus G} \frac{(|\mathcal{G}| - 1 - |S|)! \cdot |S|!}{|\mathcal{G}|!} \Delta_G(S), \quad (15)$$

which is a weighted average of marginal contributions to all possible combinations of groups.

The GSI cannot always be calculated in a time-efficient way, because the number of coalitions $S \subset \mathcal{G} \setminus G$ can become large very quickly. In practice, the Shapley value is often approximated (Casalicchio et al. 2019; Covert et al. 2020) by drawing $M \leq |\mathcal{G}|!$ different coalitions $S \subset \mathcal{G} \setminus G$ and averaging the marginal, weighted contributions:

$$\hat{\phi}_M(G) = \frac{1}{M} \sum_{m=1}^M (|\mathcal{G}| - 1 - |S_m|)! \cdot |S_m|! \cdot \Delta_G(S_m), \quad (16)$$

with $S_m \subset \mathcal{G} \setminus G$, for all $m = 1, \dots, M$.

The GSI can in general not be exactly decomposed into the sum of the Shapley importances for single features of the regarded group. In Appendix B, we show that the remainder term $R = \phi(G) - \sum_{i \in G} \phi(x_i)$ depends only on higher-order interaction effects between features of the regarded group and features of other groups. Hence, if one is interested in which features contributed most within a group, the Shapley importances for single features can be calculated, which provide a fair distribution of feature interactions within the group but not necessarily of feature interactions across groups. However, the remainder term can be used as a quantification of learned higher-order interaction effects between features of different groups.

While the GSI can be calculated with permutation- as well as refitting-based approaches, we will only apply the permutation-based approach in the upcoming simulation studies and the real-world example.

3.4 Real world use cases

For each dataset from Sect. 1.1, we fitted a random forest and summarized the three most important groups according to different grouped feature importance methods. For the importance scores of LOGI and LOGO, we used a 10-fold cross-validation (Table 2).

For the *birthweight* task, the feature **lwt** (mother's weight) was the most important group to predict the birthweight for all grouped feature importance methods except for LOGI. While all methods except LOGI also agree on the second most important group **ui** (presence of uterine irritability), feature groups differ for the third rank. However, this may also be due to statistical variability, as the importance values become very

Table 2 Best 3 groups for each grouped feature importance score

Dataset	GPMI	GPMI	GSI	LOGI	LOGO
<i>Birthweight</i>	lwt (0.067)	lwt (0.056)	lwt (0.062)	ui (0.041)	lwt (0.036)
	ui (0.056)	ui (0.047)	ui (0.046)	Race (0.017)	ui (0.029)
	Smoke (0.009)	Race (0.045)	ptl (0.019)	ptl (0.015)	Race (0.005)
<i>Colon</i>	Gene14 (0.143)	Gene14 (0.174)	Gene14 (0.125)	Gene14 (0.128)	Gene14 (0.131)
	Gene10 (0.007)	Gene16 (0.087)	Gene16 (0.042)	Gene20 (0.045)	Gene17 (0.036)
	Gene7 (0.001)	Gene12 (0.057)	Gene13 (0.019)	Gene13 (0.028)	Gene18 (0.033)
<i>Fuelsubset</i>	NIR (30.51)	NIR (42.20)	NIR (36.21)	NIR (27.35)	NIR (8.34)
	UVVIS (2.85)	UVVIS (14.38)	UVVIS (7.99)	UVVIS (15.74)	H ₂ O (0.14)
	H ₂ O (0.01)	H ₂ O (1.26)	H ₂ O (0.24)	H ₂ O (-12.17)	UVVIS (-2.14)

For the classification task (*colon*) the scores were calculated as differences in classification accuracy. For the other two regression tasks the scores result from differences in MSE

small. It is interesting that **lwt**, despite being the most important group for all other scores, is not very important in terms of LOGI. Thus, **lwt** is less important as a stand-alone group, but appears important if the other feature groups are included in the model.

In the *colon* task, the feature group *gene14* is by far the most important group to predict cancerous tissue for all grouped feature important methods. However, there are variations in the second and third most important groups.

For the *fuelsubset* task, the permutation-based grouped importance methods (GPMI, GPMI and GSI) show the same importance ranking for the three most important feature groups. However, for the refitting-based grouped importance methods (LOGI and LOGO), we can observe interesting differences. The features from the *UVVIS* group are important as a stand-alone group as can be seen by their positive LOGI score. However, the negative LOGO score of the *UVVIS* group indicates that the algorithm seems to perform better with only the *NIR* and *H₂O* groups.

GPMI, GPMI and GSI provide importance scores for feature groups of a given trained model without the necessity to refit the model. In contrast, LOGI and LOGO provide grouped importance scores based on the underlying algorithm and should always be considered together.

4 Sequential grouped feature importance

In general, feature groups do not necessarily have to be distinct or independent of each other. When groups partly contain the same or highly correlated features, we may obtain high grouped feature importance scores for similar groups. This can lead to misleading conclusions regarding the importance of groups. Quantifying the importance of different combinations of groups is especially relevant in applications where extra costs are associated with using additional features from other data sources. In this case, one might be interested in the sparsest, yet most important combination of groups or in understanding the interplay of different combinations of groups. Hence,

in practical settings, it is often important to decide which additional group of features to make available (e.g., buy or implement) for modeling and how groups should be prioritized under economic considerations.

Gregorutti et al. (2015) introduced a method called *grouped variable selection*, which is an adaptation of the recursive feature elimination algorithm from Guyon et al. (2002) and uses permutation-based grouped feature importance scores for the selection of feature groups. In Algorithm 1, we introduce a sequential procedure that is based on the idea of stability selection (Meinshausen and Bühlmann 2010). The procedure primarily aims at understanding the interplay of different combinations of groups by analyzing how the importance scores change after including other groups in a sequential manner. The feature groups must be pre-specified by the user. We prefer a refitting-based over a permutation-based grouped feature importance score when the secondary goal is to find well-performing combinations of groups. Here, the fundamental idea is to start with an empty set of features and to sequentially add the next best group in terms of LOGI until no further substantial improvement can be achieved. Our sequential procedure is based on a greedy forward search and creates an implicit ranking by showing the order in which feature groups are added to the model. To account for the variability introduced by the model, we propose to use repeated subsampling or bootstrap with sufficient repetitions (e.g., 100 repetitions).

To better understand Algorithm 1, we will demonstrate it with a small example with four groups $\mathcal{G} = \{G_1, G_2, G_3, G_4\}$ here. As a reminder, each group is a subset of $\{1, \dots, p\}$, and we want to find a subset $B \subset \{1, \dots, p\}$, which consists of the union of groups in \mathcal{G} . The subset B is found by our sequential grouped feature importance procedure. To account for variability, the whole dataset is split into two sets (training and test set) repeatedly so that the train-test splits are different in each repetition of the resampling strategy (bootstrap or subsampling). For each training set, Algorithm 1 starts with an empty set $B = \emptyset$ (line 2, Algorithm 1). In line 5 of Algorithm 1, the candidate set $\mathcal{B} \subset \mathcal{P}(\mathcal{G})$ is defined as all subsets of the power set with cardinality 1. These are all individual groups $\mathcal{B} = \{\{G_1\}, \{G_2\}, \{G_3\}, \{G_4\}\}$. The LOGI score of each single group is then calculated. In our example, let G_1 have the highest LOGI score, which also exceeds the threshold δ . The desired combination B is preliminarily defined as G_1 (line 8), and for the comparison in the next step, the LOGI score of G_1 is defined as L_0 (line 9). Then, a new candidate set \mathcal{B} is defined (line 11), which consists of all subsets of the power set of \mathcal{G} of size i (at this step, we have $i = 2$), where $B = G_1$ is also a subset of \mathcal{B} . Hence, $\mathcal{B} := \{\{G_1, G_2\}, \{G_1, G_3\}, \{G_1, G_4\}\}$. The LOGI score of elements of \mathcal{B} is calculated as the LOGI score of the union of all subsets. Now, let $\widehat{LOGI}(G_1 \cup G_3)$ have the highest score. This score is compared to the LOGI score of the previous iteration L_0 (line 13). Let the difference exceed the threshold δ for our example. In line 14 and 15, the desired combination B is now defined as $G_1 \cup G_3$ and the LOGI score is again defined as L_1 . Algorithm 1 now jumps to line 10 again with $i = 3$. The candidate set is now $\mathcal{B} = \{\{G_1, G_3, G_2\}, \{G_1, G_3, G_4\}\}$ (line 11). The LOGI scores are now calculated again for each element of \mathcal{B} . Let no LOGI score exceed L_0 by the threshold δ (line 13). Algorithm 1 now ends for this dataset split and returns $B = G_1 \cup G_3$ as the best combination. This procedure is repeated for each train-test split in each repetition.

Algorithm 1: Sequential Grouped Feature Importance

```

input : Set of groups  $\mathcal{G} = \{G_1, \dots, G_k\}$ .
         Improvement threshold  $\delta > 0$ .
         Number of repetitions for the data splitting.
output: For every data split: a combination  $B \subset \{1, \dots, p\}$  and the order in which feature groups
         were added.
1 for Every outer data split do
2   Let  $B = \emptyset$  for  $i = 1, \dots, k$  do
3     if  $i = 1$  then
4       Define candidate set  $\tilde{B} := \{\tilde{G} \in \mathcal{P}(\mathcal{G}) \mid |\tilde{G}| = 1\}$ 
5       Find best single group  $G^* = \arg \max_{\tilde{G} \in \tilde{B}} (\widehat{LOGI}(\tilde{G}))$ 
6       if  $\widehat{LOGI}(G^*) > \delta$  then
7          $B = G^*$ 
8          $L_{i-1} = \widehat{LOGI}(B)$ 
9
10      if  $i > 1$  and  $B \neq \emptyset$  then
11        Define candidate set  $\tilde{B} := \{\tilde{G} \in \mathcal{P}(\mathcal{G}) \mid |\tilde{G}| = i \text{ and } B \subset \tilde{G}\}$ 
12        Find best combination  $G^* = \arg \max_{\tilde{G} \in \tilde{B}} (\widehat{LOGI}(\bigcup_{G' \in \tilde{G}} G'))$ 
13        if  $\widehat{LOGI}(\bigcup_{G' \in G^*} G') - L_{i-1} > \delta$  then
14           $B = \bigcup_{G' \in G^*} G'$ 
15           $L_{i-1} = \widehat{LOGI}(B)$ 
16        else
17          break for loop
18

```

Since the order in which feature groups are added is also known, alluvial charts (Allaire et al. 2017) can be created for visualization purposes (see Figs. 2 and 10). In these charts, we included the number of times feature groups were added as well as the performance on the test datasets. These charts show how frequently a group was selected given that another group was already included and thereby highlight robust combinations of groups.

5 Comparison of grouped feature importance methods

After introducing the methodological background of the different loss-based grouped feature importance measures in Sect. 3, we will now compare them in different simulation settings. We analyze the impact on all methods for settings where (1) groups are dependent, (2) correlations within groups vary, and (3) group sizes differ.

5.1 Dependencies between groups and sparsity

In this section, we compare refitting- and permutation-based grouped feature importance methods and show how different dependencies between groups can influence the importance scores. We demonstrate the benefits of the sequential grouped feature importance procedure and conclude with a recommendation of when to use refitting or permutation-based methods depending on the use-case.

We simulate a data matrix \mathbf{X} with $n = 1000$ instances and 3 groups G_1, G_2, G_3 , with each of them containing 10 normally distributed features. Features are simulated in such a way that features within each group are highly correlated. However, features in G_3 are independent of features in G_1 and G_2 , while features in G_1 and G_2 are also highly correlated with each other. To generate normally distributed features with such correlation patterns, we follow the approach of Toloşi and Lengauer (2011) and use prototype vectors in the following way: (1) We draw n instances of the prototype vector $\mathbf{U} \sim \mathcal{N}(0, 1)$. (2) We generate features in G_1 by adding a normally distributed error term $\epsilon \sim \mathcal{N}(0, 0.5)$ to 10% of the instances of the prototype vector \mathbf{U} . (3) Features in G_2 are generated by copying features of G_1 and adding a small normally distributed error term $\epsilon \sim \mathcal{N}(0, 0.01)$ to the copied features. It follows that features within G_1 and G_2 as well as features between the two groups are highly correlated. (4) We generate a new prototype vector \mathbf{V} , which is independent of \mathbf{U} . (5) We generate features for G_3 in the same way as done for G_1 in step (2) but with the prototype vector \mathbf{V} .

The target vector \mathbf{Y} is generated by $\mathbf{Y} = 2\mathbf{U} + \mathbf{V} + \epsilon$, with $\epsilon \sim \mathcal{N}(0, 0.1)$. We fitted a support vector machine with a radial basis function kernel⁵, as an example of a black-box algorithm.

The results in Table 3 show that there can be major differences depending on how the grouped feature importance is calculated. Permutation methods (GOPFI & GPFI & GSI) reflect the importance of the groups based on a model trained on a fixed dataset. In contrast, refitting methods (LOGI & LOGO) retrain the model on a reduced dataset and can therefore learn new relationships. Looking at the results from the permutation methods, we can see that the groups G_1 and G_2 are approximately equally important while both being more important than G_3 . However, the results from the refitting methods can reveal some interesting relationships between the groups. The refitting methods highlight that G_1 and G_2 are more or less interchangeable if we only consider a performance-based interpretation (which might not coincide with a domain-specific

⁵ Epsilon regression, $\epsilon = 0.1$, $C = 1$ with heuristically chosen kernel width according to (Caputo et al. 2002) (here: $\sigma = 0.079$).

Table 3 Results of different feature importance calculations of the simulation

Group	GOPFI	GPMI	GSI	LOGI	LOGO
G_1	6.04 (± 0.37)	2.64 (± 0.07)	4.12 (± 0.45)	3.93 (± 0.75)	-0.01 (± 0.02)
G_2	5.90 (± 0.35)	2.57 (± 0.09)	4.01 (± 0.47)	3.93 (± 0.76)	-0.00 (± 0.02)
G_3	1.76 (± 0.39)	1.75 (± 0.05)	1.54 (± 0.39)	0.58 (± 1.01)	1.01 (± 0.22)

GSI scores were calculated without approximation, with v_{perm} as value function (see Eq. 14). All results were averaged by a 10-fold cross-validation scheme, with standard deviations reported in parentheses

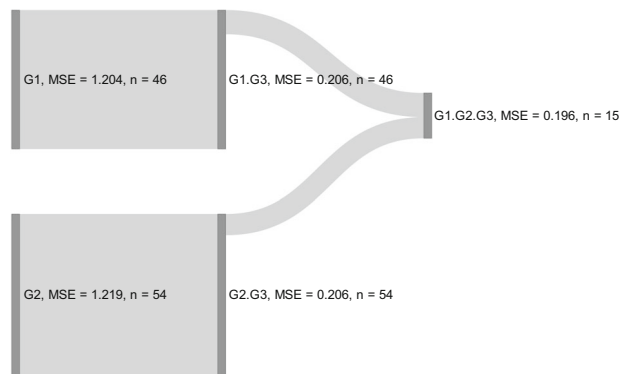


Fig. 2 Sequential grouped feature importance for the simulation in Sect. 5.1. 100 times repeated subsampling. Improvement threshold $\delta = 0.001$. Vertical bars show one step of the sequential procedure (left to right). Height of the vertical bars represent the number of subsampling iterations that a combination of groups was chosen. *MSE* scores show predictive performance. Streams represent the addition of a group

perspective)⁶. Hence, the two groups do not complement each other. This is reflected by the near-zero LOGO scores, which indicate that leaving each group out of the full model does not considerably change the model's expected loss.

Figure 2 illustrates the results of the sequential procedure introduced in Algorithm 1. We see that across 100 subsampling iterations, G_1 was chosen 46 times as the most important first group, and G_2 was chosen 54 times with similar predictive performance for both groups, while G_3 was never chosen as the first most important group. Hence, similar to LOGI, we can see that if only one group can be chosen, it would either be G_1 or G_2 with approximately the same probability. In the second step, the group G_3 was added in all cases to either G_1 or G_2 (depending on which group had been chosen in the first step). This step resulted in an on-average drop in the MSE score from 1.2 to 0.2. In only a few cases (15 out of 100), the final addition of either G_1 or G_2 to a full model in step 3 exceeded the very low chosen threshold of $\delta = 0.001$. This rather unlikely improvement is represented by the proportionally narrower band that connects the second and the third step (dark gray bars) in the chart in Fig. 2. This reveals that these two groups are—from a performance or loss perspective—rather interchangeable and do not benefit from one another.

⁶ It is possible that adding a group of features to the model might not lead to a better model performance, but the group may still be relevant due to the domain-specific context. However, this depends on the regarded use case. All our interpretations here are purely statistical.

The choice between using permutation-based or refitting-based grouped feature importance methods might depend on the number of groups and correlation strength between the different groups. If feature groups are distinct and features between the groups are almost uncorrelated, we might prefer permutation over refitting methods due to lower computation time. In cases where groups are correlated with each other (e.g., because some features belong to multiple groups), refitting methods might be preferable, as they are not misleading in correlated settings. Since the number of groups is usually smaller than the number of features in a dataset, refitting methods for groups of features could become a viable choice. Furthermore, with the sequential grouped feature importance procedure, it is possible to find sparse and well performing combinations of groups in an interpretable manner. Thus, this approach helps to better understand which groups of features were important (e.g., as they were more frequently selected) given that certain groups were already selected.

5.2 Varying correlations within groups

In many use cases, it is quite common to group similar (and therefore, often correlated) features together, while groups of features may be almost independent of each other. However, compared to Sect. 5.1, correlations of features within groups might differ. We created a data matrix \mathbf{X} with $n = 1000$ instances and 4 groups G_1, G_2, G_3 , and G_4 , with each of these groups containing 10 normally distributed features. Using fivefold cross-validation, we fitted a random forest with 2000 trees and a support vector regression with a radial basis function kernel.⁷ The univariate target vector \mathbf{Y} is defined as follows:

$$\mathbf{Z}_j = 3\mathbf{X}_{G_j,3}^2 - 4\mathbf{X}_{G_j,5} - 6\mathbf{X}_{G_j,7} + 5\mathbf{X}_{G_j,9} \cdot d_j, \quad j \in \{1, 2, 3\}$$

$$\mathbf{Y} = \sum_{j=1}^3 \mathbf{Z}_j + \epsilon$$

with

$$d_j = \begin{cases} 1, & \text{if } \text{mean}(\mathbf{X}_{G_j,8}) > 0 \\ 0, & \text{otherwise} \end{cases}$$

and $\epsilon \stackrel{iid}{\sim} N(0, 1)$. The i -th feature of the j -th group is denoted by $\mathbf{X}_{G_j,i}$. We repeated the simulation 500 times.

It follows that G_1, G_2 , and G_3 have the same influence on the target variable, while G_4 has no influence on \mathbf{Y} . We generate the feature space \mathbf{X} —similar to the approach in Sect. 5.1—as follows: (1) For each feature group j , we generate a prototype vector $\mathbf{U}_j \sim \mathcal{N}(0, 1)$ with n instances. (2) We generate the features of a group G_j by altering a proportion α with $0 \leq \alpha \leq 1$ of the n instances of \mathbf{U}_j . We alter these instances by taking

⁷ We used a cost parameter of $C = 1$ and estimate the kernel width based on the heuristic introduced by Caputo et al. (2002)

a weighted average between the respective values of U_j (20%) and a standard normally distributed random variable W_i (80%). For the results shown in Fig. 3, we set α to 0.1 for all features within the same group. Hence, correlations within groups are the same (around 90%) for all groups, while groups themselves are independent of each other. The plots show that all methods correctly attribute the same importance to the first three groups, while the fourth group is not important for predicting Y . The lower plots in Fig. 3, on the other hand, correlations within groups vary across groups. The altering proportion parameter α is set to 0.1 for features of G_1 and G_4 , to 0.3 for features of G_2 , and to 0.6 for features of G_3 . Hence, features in G_1 and G_4 are highly correlated within the respective group, while features within G_2 and G_3 show a medium and small correlation, respectively. While G_4 is still recognized to be unimportant, the relative importance of groups 1 to 3 drops with decreasing within-group correlation. This artifact seems—at least, in this simulation setting—to be even more severe for the random forest compared to the support vector machine. For example, G_3 is on average less than half as important as G_1 for permutation-based methods. Thus, none of the methods reflect the true importance of the different groups of the underlying data generating process. A possible reason for this artifact is that the regarded model learned effects different from those given by the underlying true relationship. Especially for the random forest, this has already been studied extensively in the presence of different correlation patterns in the feature space (Strobl et al. 2008; Nicodemus et al. 2010). Additionally, Hooker and Mentch (2019) showed that permutation-based methods are more sensitive in this case than refitting methods, which is also visible for both models in Fig. 3. Since the model is learned on the original feature space and group structures are not considered in the modelling process, we can also observe this effect when applying grouped feature importance methods. This is due to the fact that we can only quantify which groups are important for the model or algorithm performance but not for the underlying data generating process, which is usually unknown. Another approach to quantify feature importance when using random forests is to extract the information on how often a feature has been used as a splitting variable for the different trees. The feature chosen for the first split has the most influence within each tree. Hence, we calculated for each repetition the percentage of how often a feature is chosen as the first splitting feature. The distribution over all repetitions is displayed in Fig. 4. Each of the features of G_1 is on average chosen more often as the first splitting feature than all features of the other groups, no matter if it has an influence on the target or not. The influential features of G_3 (which has the lowest within-group correlation) are rarely chosen as the first splitting feature. This observation confirms the results of the grouped importance methods in Fig. 3, since all of them rank G_3 as least important from the influential feature groups.

Note that while GPFI and LOGO are calculated with reference to the full model's performance—which on average leads to higher absolute values than the two counter-methods based on the null model's performance—GOPFI and LOGI might lead to less robust results, as the newly learned effects as well as the approximation of the permutation effect underlie a higher uncertainty. This effect might increase when relative values instead of absolute values are considered due to smaller absolute importance scores of GOPFI and LOGI. However, the methods are only comparable on a relative scale. This effect is also visible in the boxplots of Fig. 3. Furthermore, LOGI can also

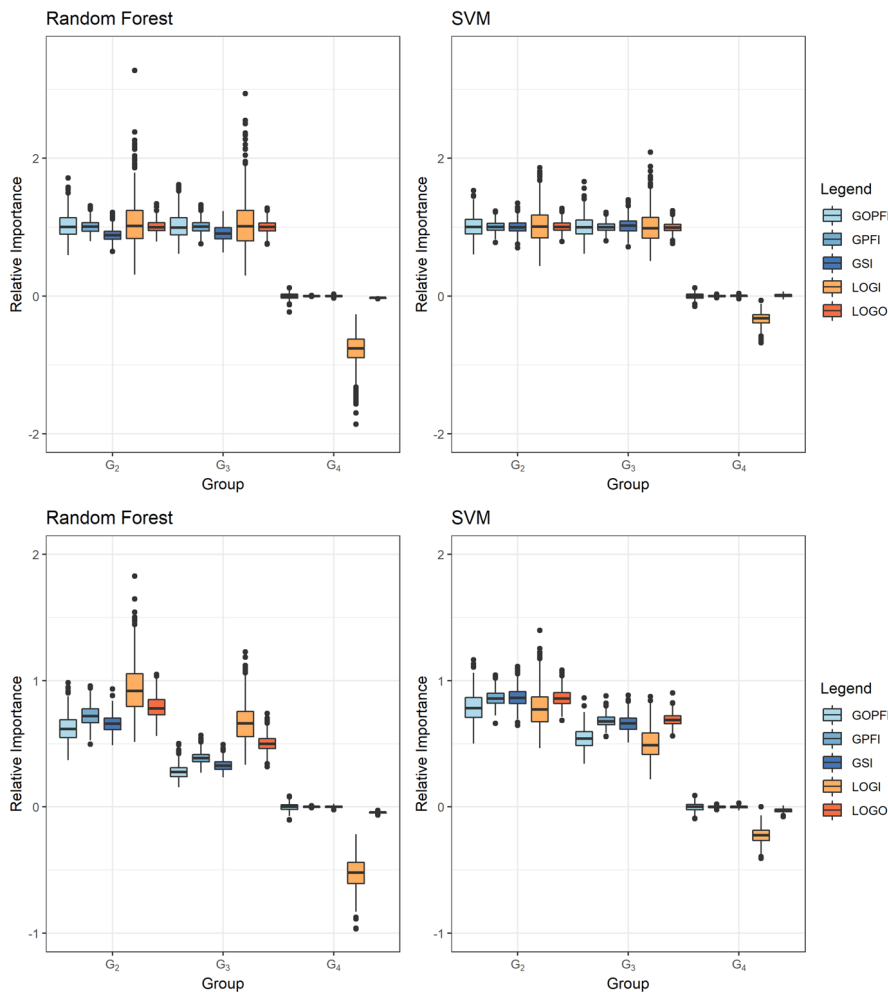


Fig. 3 Upper (lower) plots: Grouped relative importance scores in the case of *equally sized* (*varying sizes*) within-group correlations for random forest (left) and SVM (right). Relative importance is calculated by dividing each of the absolute group importance scores by the importance score of G_1 . Hence, the relative importance of G_1 is 1. Boxplots illustrate the variation between different repetitions

take negative values in the case of G_4 , as the feature group does not affect the target in the underlying data generating process, and hence it might be counterproductive to only include G_4 compared to the null model.

5.3 Varying sizes of groups

Another factor to consider when calculating grouped rather than individual feature importance scores is that differing group sizes might influence the ranking of the scores. Groups with more features might often have higher grouped importance scores and

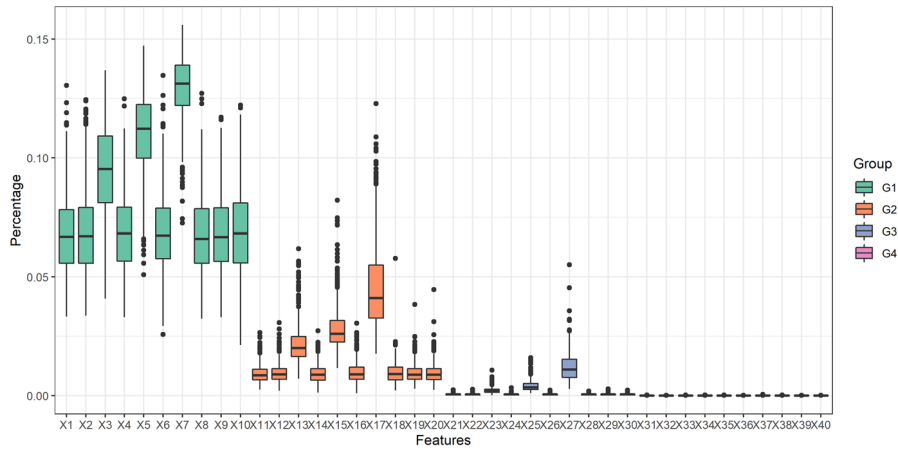


Fig. 4 Percentage of how often each feature is chosen as the first splitting feature within the trained random forests. Results have been averaged over the cross-validation folds for each repetition. Boxplots show the distribution over all 500 repetitions

might contain more noise features than smaller groups. Therefore, Gregorutti et al. (2015) argue that in case one must decide between two groups that have an equal importance score, one would prefer the group with fewer features. Following from that, they normalize the grouped feature importance scores regarding the group size with the factor $|G|^{-1}$. This is also used in the default definition of the grouped model reliance score in Valentin et al. (2020). However, the usefulness of normalization highly depends on the question the user would like to answer. This is illustrated in a simulation example in Fig. 5. We created a data matrix \mathbf{X} with $n = 2000$ instances and 2 groups, with G_1 containing $\{x_1, \dots, x_6\}$ and G_2 containing $\{x_7, x_8\}$ i.i.d. uniformly distributed features on the interval $[0, 1]$. The univariate target variable \mathbf{Y} is defined as follows:

$$\mathbf{Y} = 2\mathbf{X}_1 + 2\mathbf{X}_3 + 2\mathbf{X}_7 + \epsilon, \quad \text{with } \epsilon \stackrel{iid}{\sim} N(0, 1).$$

We used 1000 observations for fitting a random forest with 2000 trees and 1000 observations for prediction and calculating the GSI as defined in Sect. 3.3 with a permutation-based value function. This was repeated 500 times. Figure 5 shows that G_1 is about twice as important as G_2 . As shown in Sect. 3.3 and Appendix B, we can compare the GSI with the Shapley importance on feature level. In case there are no higher-order interaction terms between groups modeled by the random forest, the single feature importance scores will approximately sum up to the grouped importance score, as shown in this example. This provides a more detailed view of how many and which features are important within each group. In this case, there are two equally important features in G_1 and one equally important feature in G_2 . If we use the normalization constant in this example, we would divide the grouped importance score of G_1 (which is on average approximately 1.1) by 6 and the one of G_2 (which is on average approximately 0.55) by 2. Consequently, G_2 with a normalized score of

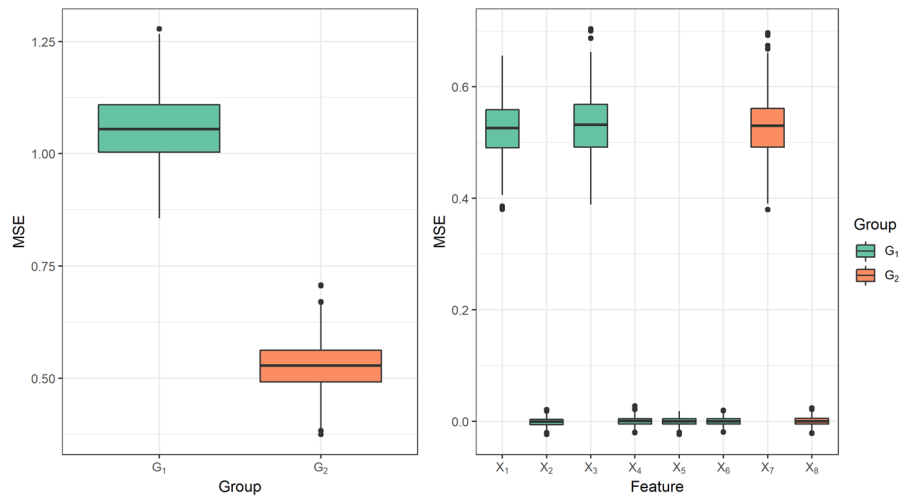


Fig. 5 Shapley importance on group (left) and on feature level (right). Boxplots show the variation between the 500 repetitions of the experiment

approximately 0.27 would be regarded as more important than G_1 with a normalized score of approximately 0.18. It follows that if we must decide between two groups, we would choose G_2 when we follow the approach of Gregorutti et al. (2015). However, since G_1 contains two features with the same importance as the one important feature of G_2 , and hence G_1 contains more information from a statistical perspective, the user might prefer G_1 . Furthermore, breaking down the GSI to the single-feature Shapley importance scores puts the user in the position of defining sparser groups by excluding non-influential features.

Finally, Table 4 presents a summary of the key takeaways regarding all discussed grouped feature importance methods.

6 Feature effects for groups

Feature effect methods quantify or visualize the influence of features on the model's prediction. For a linear regression model, we can easily summarize the feature effect in one number, thus making interpretation very simple: If we change feature x_1 by one unit, our prediction will change by the corresponding coefficient estimate $\hat{\beta}_1$ (positively or negatively depending on the sign of the coefficient). For more complex non-linear models like generalized additive models, such a simplified summary of the feature effect is not adequate, as the magnitude and sign of the effect might change over the feature's value range. Hence, it is more common to visualize the marginal effect of the feature of interest on the predicted outcome. Since ML models are often complex non-linear models, different visualization techniques for the feature effect have been introduced in recent years. Common methods are PDP, ICE curves or ALE (Friedman 2001; Goldstein et al. 2013; Apley and Zhu 2019), which show how changes in the feature values affect the predictions of the model. However, these are

Table 4 Overview of pros and cons of the grouped feature importance methods

Criteria	GOPFI & GPFI	GSI	LOGI & LOGO
Time efficient	Yes (in comparison to alternatives)	Depends on number of groups	Depends on number of groups
Dependencies between groups (Sect. 5.1)	No full picture	No full picture	More insights than permutation-based if regarded together
Identify well performing combinations of groups (Sect. 5.1)	Not in general	Not in general	Only LOGI within Algorithm 1
Correlations within groups but independence between groups (Sect. 5.2)	Depends on learned effects of the model, less problematic if within group correlations do not differ strongly between groups	Depends on learned effects of the model, less problematic if within group correlations do not differ strongly between groups	More robust than permutation-based methods but still dependent on learned effects
Drilldown of grouped importance score on feature level (Sect. 5.3)	No	Yes (approximately depending on the influence of higher-order interactions)	No

While GOPFI is less relevant on its own, LOGI can provide insightful interpretations, e.g., if feature groups are correlated with each other or when used within the sequential procedure introduced in Sect. 4. The sequential procedure is the only method that can identify well performing and sparse combination of groups. Note that GSI is only evaluated w.r.t. a permutation-based calculation

usually only defined for a maximum of two features. For larger groups of features, this becomes more challenging, since it is difficult to visualize the influence of several features simultaneously. The approach described in this section aims to create effect plots for a predefined group of features that have an interpretation similar to that of the single-feature PDP. To achieve this, we transform the high-dimensional space of the feature group into a low-dimensional space by using a supervised dimension reduction method, which is discussed in Sect. 6.1. We want to find a few underlying factors that are attributed to a sparse and interpretable combination of features that explain the effect of the regarded group on the model's expected loss. We provide a detailed description of this method in Sect. 6.3 and introduce the resulting combined features effect plot (CFEP). In Sect. 6.4, we illustrate the advantages of applying a supervised rather than an unsupervised dimension reduction method and compare our method to the main competitor, which is the totalvis effect plot introduced in Seedorff and Brown (2021).

6.1 Choice of dimension reduction method

The most prominent dimension reduction technique is arguably PCA (Jolliffe 1986). PCA is restricted to explaining most of the variance of the feature space, and the identified projections are not related to the target variable (for more details see Appendix

C.1). Because we want to visualize the mean prediction of combined features as a result of the dimension reduction process, we prefer supervised procedures that maximize dependencies between the projected data \mathbf{XV} —with \mathbf{V} being a projection $\mathbf{V} \in \mathbb{R}^{p \times p}$ —and the target vector \mathbf{Y} (as we show in Sect. 6.4). Many methods for supervised PCA have been established. For example, see Bair et al. (2006), who used a subset of features that were selected based on their linear correlation with the target variable. Another very popular method that maximizes the covariance between features and the target variable is partial least squares (PLS) (Wold et al. 1984). The main difference between these methods and the supervised PCA (SPCA) introduced by Barshan et al. (2011) is that the SPCA is based on a more general measure of dependence, called the Hilbert-Schmidt Independence Criterion (HSIC). This independence measure is constructed to be zero, if and only if any bounded continuous function between the feature and target space is uncorrelated. In practice, an empirical version of the HSIC criterion is calculated with kernel matrices. It follows that while this SPCA technique can cover a variety of linear and non-linear dependencies between \mathbf{X} and \mathbf{Y} by choosing an appropriate kernel, the other suggested methods are only able to model linear dependencies between the features and the target variable. The approach that is probably best suited for our application of finding *interpretable* sets of features in a high-dimensional dataset is the method called sparse SPCA, described in Sharifzadeh et al. (2017). Similar to the SPCA method from Barshan et al. (2011), sparse SPCA not only uses the HSIC criterion to maximize the dependency between projected data \mathbf{XV} and the target \mathbf{Y} , but also incorporates an L_1 penalty of the projection \mathbf{V} for sparsity. The sparse SPCA problem can be solved with a *penalized matrix decomposition* (Witten et al. 2009). More theoretical details on the sparse SPCA, including the HSIC criterion and how it can be calculated empirically, and the choice of kernels and hyperparameters can be found in Appendix C.

6.2 Totalvis effect plot

Seedorff and Brown (2021) recently introduced a method that aims to plot the combined effect of multiple features by using PCA. Their approach can be described as follows: First, they apply PCA on the regarded feature space to receive the principal components matrix after rotation. For the principal component of interest, they create an equidistant grid. Second, for each grid value, they replace all values of the selected principal component with this grid value and transform the matrix back to the original feature space. Third, The ML model is applied on these feature values and a mean prediction for the grid point of the regarded principal component is calculated. Steps 2 and 3 are repeated for all grid points.

Hence, with this method, combined effect plots for up to p principal components can be created. Thus, Seedorff and Brown (2021) do not focus on explaining groups of features explicitly. Furthermore, they use PCA for unsupervised dimension reduction, and thus, projections might not be related to the target. Due to using PCA and not sparse PCA, the results might be difficult to interpret, as many or all features might have an influence on the principal component. Lastly, with the back-transformation from the principal component matrix to the original feature space, all feature values

change and might not be meaningful anymore. For example, in the case of integer features, the back-transformation might lead to real feature values. We illustrate the drawbacks of the method compared to the CFEP in Sect. 6.4.

6.3 Combined features effect plot (CFEP)

The CFEP picks up the idea of PDPs (Friedman 2001) and extends it to groups of features. The partial dependence function is defined as

$$f_S^{PD}(\mathbf{x}_S) = \mathbb{E}_{X_C}[\hat{f}(\mathbf{x}_S, X_C)] \quad (17)$$

with $S \subset \{1, \dots, p\}$ and $C = \{1, \dots, p\} \setminus S$. Since the joint distribution of X_C is usually unknown, the Monte Carlo method is used to estimate $f_S^{PD}(\mathbf{x}_S)$:

$$\hat{f}_S^{PD}(\mathbf{x}_S) = \frac{1}{n} \sum_{i=1}^n \hat{f}(\mathbf{x}_S, \mathbf{x}_C^{(i)}) \quad (18)$$

Hence, we marginalize over all features in C and with that we obtain the average marginal effect for the feature subset in S . The PDP usually visualizes this average marginal effect for $|S| \leq 2$ by plotting $(\mathbf{x}_S^{(k)}, \hat{f}_S^{PD}(\mathbf{x}_S^{(k)}))$ for some pre-specified grid points $k = \{1, \dots, m\}$.⁸ However, this is usually only possible for $|S| \leq 2$ and thus not directly applicable to visualize the combined effect of feature groups. To obtain a visualization in the case of $|S| > 2$, we need to reduce the dimensions and therefore define the CFEP of a certain group of features G as follows:

- (1) We first apply a suitable (preferably supervised) dimension reduction method (e.g., here we use the sparse SPCA, however, the CFEP follows a modular approach and hence the dimension reduction method is exchangeable) on features in $G \subset \{1, \dots, p\}$ to obtain a low dimensional representation of the feature group G . We denote these principle component functions—which are ordered according to relevance⁹ and which possibly depend on a reduced set of features¹⁰ $S_j \subseteq G$ with $j \in \{1, \dots, |G|\}$ —by $g_j : \mathcal{X}_{S_j} \rightarrow \mathbb{R}$.
- (2) For visualization purposes, we choose from all possible g_j with $j \in \{1, \dots, |G|\}$ a principle component function

$$g : \mathcal{X}_S \rightarrow \mathbb{R} \quad (19)$$

(with S being its reduced set of features) which serves as a proxy for the feature group G . We usually only consider the first few principle components.

⁸ For example, by using an equidistant grid or a random sample of values of \mathbf{x}_S .

⁹ The relevance is defined by the objective that is optimized by the dimension reduction method. For sparse SPCA this is the HSIC criterion (see also Appendix C) and for PCA it is the explained variance.

¹⁰ If a dimension reduction method which results in a sparse solution (e.g., sparse SPCA) is applied, then S_j is only a subset of G and might differ for different principal components.

- (3) We calculate the average marginal effect $\hat{f}_S^{PD}(\mathbf{x}_S)$ of the feature set S exactly as in Eq. (18).
- (4) We visualize the CFEP by plotting $(g(\mathbf{x}_S^{(i)}), \hat{f}_S^{PD}(\mathbf{x}_S^{(i)}))$ for each observation in the dataset.

Hence, the CFEP visualizes the average marginal effect of features in S against the combinations of features received by the dimension reduction method (e.g., a linear combination of a principal component in the case of sparse SPCA) and thus shows how different values of $g(\mathbf{x}_S)$ affect the predictions of a given model. For a feature group, several principle components g_j and hence several CFEPs may be of interest.

The CFEP is defined in Algorithm 2, but we will demonstrate the procedure of constructing a CFEP with the illustrative example in Fig. 6. In this example, we have two predefined groups of features, where the first group contains x_1, x_2 , and x_3 , and the second group contains features x_4 and x_5 . The sparse SPCA on the first group yields a first principal component (g_1) with the loadings 0.3 for x_1 , 0.6 for x_2 and 0.5 for x_3 (step 1 to 3 of Algorithm 2). It follows that $S = \{1, 2, 3\}$ and that the low dimensional representation of interest is g_1 . For the construction of a CFEP for g_1 , mean predictions for the principal component are calculated for each observation. To calculate the mean prediction of the first observation (shown in red), we replace the values of features with non-zero loadings of g_1 of each instance in the dataset by the feature values of the first observation (step 6 in Algorithm 2). A prediction vector $\hat{\mathbf{y}}_{rep}^{(1)}$ is then calculated with the previously trained model (step 7 in Algorithm 2). The value on the y-axis for the red point in the graph below corresponds to the mean over all predictions for the first observation: $\bar{y}_{rep}^{(1)} = (0.8 + 0.2 + 0.7 + 0.6 + 0.4 + 0.3)/6 = 0.5$. The value on the x-axis is the linear projection of the first observation for the regarded principal component (step 8 and 9 in Algorithm 2). Hence, it is calculated by the weighted sum of feature values $x_1^{(i)}$ to $x_3^{(i)}$, where the weights are defined by the loadings of the respective principal component that we receive with sparse SPCA.

In contrast to PDP or totalvis effect plots, CFEP produces a point cloud instead of a curve. The CFEP is, mathematically speaking, not a function, since points on the x-axis correspond to linear projections of features within a group. A point z on the x-axis can have multiple combinations of features, which lead to z and have different mean predictions on the y-axis. However, we now have the possibility to interpret the shape of the point cloud and can draw conclusions about the behavior of the mean prediction of the model regarding a linear combination of features of interest.

6.4 Experiments on supervised versus unsupervised dimension reduction

As discussed in Sect. 6.1, PCA might be the most popular dimension reduction method. However, since PCA is unsupervised, it does not account for the dependency between the feature space and the target variable. To evaluate the degree to which this drawback influences CFEP, we examine two regression problems on simulated data. The first is defined by a single underlying factor depending on a sparse set of features, which can be represented by a single principal component. The linear combination of this feature set is also linearly correlated with the target variable. The second regression problem

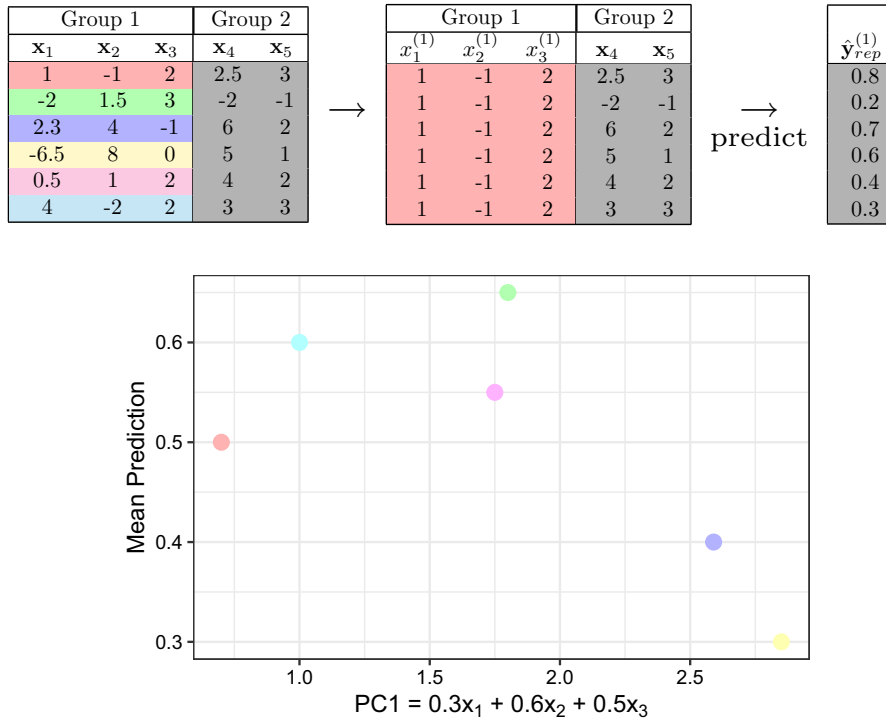


Fig. 6 Explanation of estimating and visualizing CFEP; the x-coordinate reflects the linear combination of features with non-zero loadings for g_1 , and the y-coordinate reflects the mean predictions $\hat{y}_{rep}^{(i)}$ for each observation i . The substitution of values for each observation is only done for features with non-zero loadings

contains two underlying factors that depend on two sparse sets of features. While the linear combination of the first feature set is also linearly correlated with the target, the second factor has a quadratic effect on \mathbf{Y} . In both cases, we compare the usage of sparse supervised and unsupervised PCA (sparse SPCA and sparse PCA) as dimension reduction methods within CFEP and compare them to the totalvis effect plot. Here, we investigate if the respective dimension reduction method does correctly identify the sparse set of features for each group. Additionally, we determine how accurately we can predict the true underlying relationship between the linear combination of these features and the target variable. Since we simulated the data, we know the number of underlying factors (principal components).

6.4.1 One factor

In this example, we created a data matrix \mathbf{X} with 500 instances of 50 standard normally distributed features with decreasing correlations. Therefore, all features are generated as done in Sect. 5.2. The altering proportion α is set to 0.2 for the first 10 features, to 0.4 for the next 10 features, and to 1 for the last 10 features. Thus, while the first 10 features are highly correlated with each other, the last 10 features are approximately

Algorithm 2: Combined Features Effect Plot

input : Dataset $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$,
group $G \subset \{1, \dots, p\}$,
model \hat{f} trained on \mathcal{D} .

output: Combined Features Effect Plot

- 1 Perform sparse SPCA on $\hat{\mathcal{D}} := \{(\mathbf{x}_G^{(i)}, y^{(i)})\}_{i=1}^n$;
- 2 Choose a principle component function of interest g ;
- 3 Let $S \subseteq G$ be the sparse set of features of g ;
- 4 **for** $i \in \{1, \dots, n\}$ **do**
- 5 get feature values $\mathbf{x}_S^{(i)}$;
- 6 create $\mathcal{D}_{rep}^{(i)}$ by replacing feature values from S of every observation with $\mathbf{x}_S^{(i)}$;
- 7 predict vector $\hat{\mathbf{y}}_{rep}^{(i)}$ by applying \hat{f} on $\mathcal{D}_{rep}^{(i)}$ row-wise;
- 8 calculate the mean prediction $\tilde{y}_{rep}^{(i)}$ of $\hat{\mathbf{y}}_{rep}^{(i)}$;
- 9 save $g(\mathbf{x}_S^{(i)})$ as x-coordinate and $\tilde{y}_{rep}^{(i)}$ as y-coordinate of observation i for the CFEP (see Eq. (19));

The CFEP can be used as a descriptive method to better understand the effect of a group of features on the target variable. The dimension reduction method in step 1 is exchangeable.

uncorrelated with each other. The sparse subgroup defined by the variable \mathbf{Z} is a linear combination of 5 features from \mathbf{X} and has itself a linear effect on the target variable \mathbf{Y} :

$$\mathbf{Z} = \mathbf{X}_5 - 2\mathbf{X}_8 - 4\mathbf{X}_{25} + 8\mathbf{X}_{47} + 4\mathbf{X}_{49}$$

$$\mathbf{Y} = \mathbf{Z} + \epsilon, \quad \text{with } \epsilon \stackrel{iid}{\sim} N(0, 1).$$

Hence, according to our notation, $G_{\mathbf{Z}}$ is defined by $G_{\mathbf{Z}} = \{5, 8, 25, 47, 49\}$, and thus, $X_{G_{\mathbf{Z}}}$ is the related subset of all features. We drew 100 samples and fitted a random forest with 2000 trees with each sample drawing. We used the 10-fold cross-validated results to perform sparse SPCA. For each dimension reduction method, we estimate $\hat{\mathbf{Z}}$ by summing up the (sparse) loading vector (estimated by the dimension reduction method) multiplied by the feature matrix \mathbf{X} . Therefore, $\mathbf{X}_{G_{\hat{\mathbf{Z}}}}$ is defined by the received sparse feature set. The mean prediction $\tilde{\mathbf{Y}}_{rep}$ for the CFEP is calculated as described in Sect. 6.3.

The impact of choosing a supervised over an unsupervised sparse PCA approach is shown in Fig. 7, which also shows the average linear trend and 95% confidence bands of CFEP for the simulation results. To evaluate how well the estimated mean prediction $\hat{\mathbf{Y}}_{rep}$ approximates the underlying trend, we assume that we know that \mathbf{Z} has a linear influence on the target. Thus, we fit a linear model on each simulation result. To compare the received regression lines, we evaluate each of them on a predefined grid and average over all 100 samples (represented by the red line). The confidence bands are then calculated by taking the standard deviation over all estimated regression lines on grid level and calculating the 2.5% and 97.5% quantiles using the standard normal approximation. The associated calculation steps for each of the 100 samples can be summarized as follows:

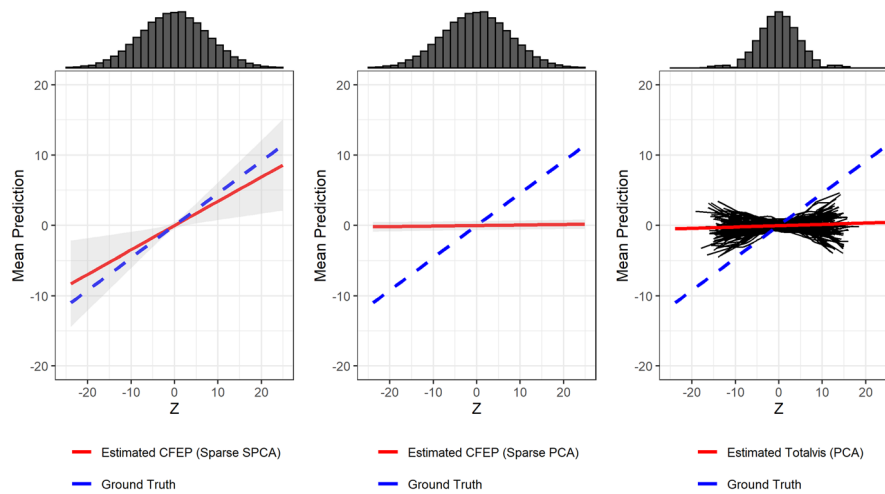


Fig. 7 Average linear trend and confidence bands of CFEP over all samples using sparse SPCA (left) and sparse PCA (middle) compared to estimated totalvis effect curves over all 100 samples for the first principal component (black) and the average linear trend (red) (right) (Color figure online)

- (1) Estimate a linear model $\hat{f}(\mathbf{X}_{G_Z}) \sim \mathbf{Z}$.
- (2) Define an equidistant grid of length 50 within the range of \mathbf{Z} .
- (3) Apply the linear model estimated in 1) on the grid defined in 2).
- (4) Repeat steps 1 to 3 for $\hat{f}(\mathbf{X}_{G_Z})$ by using the true underlying features of \mathbf{Z} to calculate the combined features dependencies that we call the ground truth.

The left plot in Fig. 7 shows a similar linear trend of the estimated CFEP compared to the average ground truth (represented by the blue line), while the red line in the right plot varies around 0. By using sparse SPCA, the underlying feature set \mathbf{X}_{G_Z} is better approximated than with sparse PCA, which is reflected in the MSE between \mathbf{Z} and $\hat{\mathbf{Z}}$ of 0.7 for sparse SPCA and 1.9 for sparse PCA. Figure 8 provides an explanation for those differences. While sparse SPCA (on average) more strongly weights features that have a large influence on the target, impactful loading weights for sparse PCA are solely distributed over highly correlated features in \mathbf{X} that explain the most variance in the feature space. Thus, including the relationship between the target and \mathbf{X} in the dimension reduction method may have a huge influence on correctly approximating the underlying factor and, hence, also on the CFEP.

Similar to using sparse PCA as a dimension reduction method within CFEP, on average, the totalvis effect curves based on PCA do not show a clear positive linear trend (see Fig. 7). For almost half of the samples, we even receive a negative instead of a positive trend for the underlying factor. The interpretation is opposite to the actual effect and, hence, is misleading.

6.4.2 Two factors

In real-world data settings are often more complex by containing non-linear relationships and the target variable is described by more than one underlying factor. Hence,

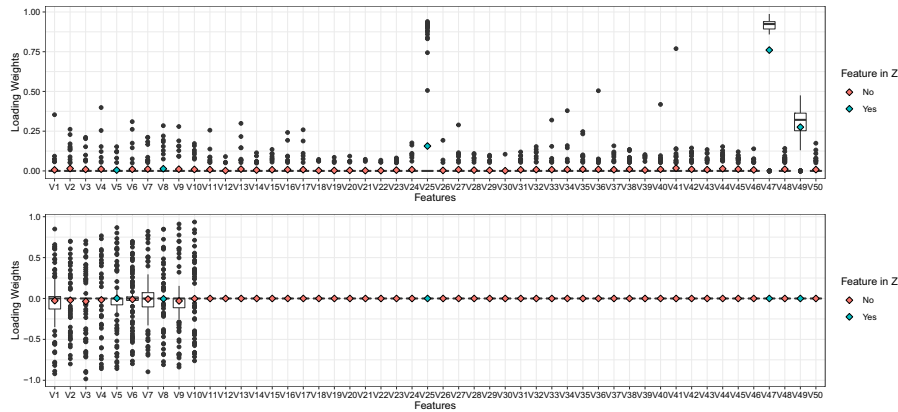


Fig. 8 Distribution of feature loadings in sparse SPCA (top) and sparse PCA (bottom) over all samples. Rhombuses denote the mean values, with the blue rhombuses indicating the features that have an influence on the target in the underlying model formula (Color figure online)

we now examine a more complex simulation setting to assess if we can observe the same behavior that we observed for the simple case. To that end, we simulated a data matrix \mathbf{X} with 500 instances for two feature sets, each containing 20 standard normally distributed features. The data for each feature set is generated as described in Sect. 5.2 but with an altering proportion of 0.15 and 0.35 for the features in the first set and 0.55 and 0.85 in the second set. Hence, within each set, the first ten features show a higher correlation among each other than the last ten features. Additionally, all features of the first set are on average more highly correlated than all features of the second set. Features between the two sets are uncorrelated. The first factor \mathbf{Z}_1 is a linear combination of four features from the first set and \mathbf{Z}_2 of two features from the second set. \mathbf{Z}_1 has a linear and \mathbf{Z}_2 a quadratic effect on \mathbf{Y} .

$$\begin{aligned} \mathbf{Z}_1 &= 3\mathbf{X}_3 - 2\mathbf{X}_8 - 4\mathbf{X}_{13} + 8\mathbf{X}_{18} \\ \mathbf{Z}_2 &= 2\mathbf{X}_{25} + 4\mathbf{X}_{35} \\ \mathbf{Y} &= \mathbf{Z}_1 + \mathbf{Z}_2^2 + \epsilon, \quad \text{with } \epsilon \stackrel{iid}{\sim} N(0, 1). \end{aligned}$$

Again, we drew 100 samples and fitted a random forest with 2000 trees with each sample drawing. The approach is almost the same as described for one factor, with the difference being that we use the first two principal components (as we want to find two sparse feature sets instead of one).

In Fig. 9, the average linear and quadratic trend of the underlying CFEPs of \mathbf{Z}_1 and \mathbf{Z}_2 are depicted for both dimension reduction methods. While the average linear regression line of sparse SPCA matches the average ground truth almost perfectly for \mathbf{Z}_1 , the associated line of sparse PCA shows only a slightly positive trend and differs substantially from the ground truth. Regarding \mathbf{Z}_2 , a similar propensity can be observed for the quadratic shape. Again, this behavior results from sparse SPCA (on

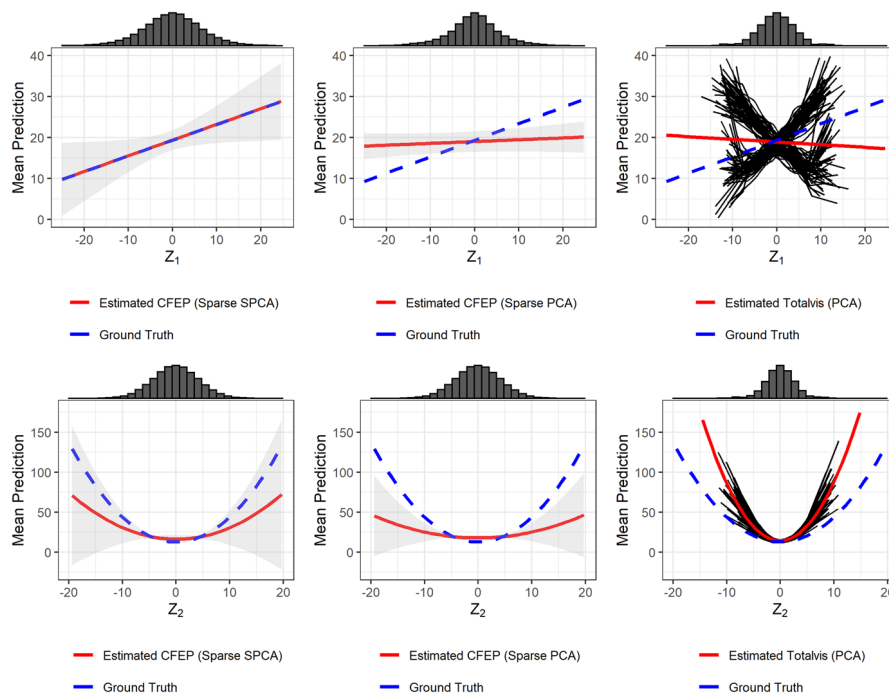


Fig. 9 Top (Z_1): Average linear trend and confidence bands of CFEP over all samples using sparse SPCA (left) and sparse PCA (middle) compared to estimated totalvis effect curves over all 100 samples for first principal component (black) and the average linear trend (red) (right). Bottom (Z_2): Same structure as for Z_1 , but showing the quadratic trend of Z_2 (Color figure online)

average) more strongly weighting features that have a large effect on the target, while the unsupervised version focuses on features that explain the most variance in \mathbf{X} .

The estimated linear trend of the totalvis effect curves for the first principal component is negative instead of positive. Thus, for most of the samples and on average, these results are completely misleading (see Fig. 9). The quadratic shape of the second component is (on average and for almost all samples) steeper than the average ground truth. Additionally, the deviation is higher here than for CFEP with sparse SPCA.

7 Real data example: smartphone sensor data

Smartphones and other consumer electronics have increasingly been used to collect data for research (Miller 2012; Raento et al. 2009). The emerging popularity of these devices for data collection is grounded in their connectivity, the number of built-in sensors, and their widespread use. Moreover, smartphones enable users to perform a wide variety of activities (e.g., communication, shopping, dating, banking, navigation, listening to music) and thus provide an ideal means to study human behavior in naturalistic contexts, over extended periods of time, and at fine granularity (Harari et al. 2015, 2016, 2017). In this regard, smartphone data has been used to investigate

individual differences in personality traits (Stachl et al. 2017; Harari et al. 2019), in human emotion and well-being (Servia-Rodríguez et al. 2017; Rachuri et al. 2010; Saeb et al. 2016; Thomée 2018; Onnela and Rauch 2016; Kolenik and Gams 2021), and in daytime and nighttime activity patterns (Schoedel et al. 2020).

We use a dataset on human behavior, collected with smartphones, to illustrate methods for group-based feature importance. The PhoneStudy dataset was consolidated from three separate datasets (Stachl et al. 2017; Schuwerk et al. 2019; Schoedel et al. 2018). It consists of 1821 features on smartphone-sensed behavior and 35 target variables on self-reported Big Five personality trait dimensions (domains) and subdimensions (facets). The dataset has been published online and is openly available.¹¹ The Big Five personality trait taxonomy is the most widely used conceptualization of stable individual differences in human patterns of thoughts, feelings, and behavior (Goldberg 1990). In their original study, Stachl et al. (2020a) used the behavioral variables to predict self-reported Big Five personality trait scores (five dimensions and 30 subdimensions) and used grouped feature importance measures to explore which classes of behaviors were most predictive for each personality trait dimension. The groups in this study were created based on theoretical considerations from past work.

The personality prediction task is challenging because (1) the dataset contains many variables on similar behaviors, (2) these variables are often correlated, and (3) effects with the targets are interactive, very small, and partially non-linear. Many variables in the dataset can be manually grouped into classes of behavior (e.g., communication and social activity, app-usage, music consumption, overall phone activity, mobility).

We use this dataset to illustrate the idea of grouped feature importance with regard to the prediction of personality trait scores for the dimension of conscientiousness (Table 5). Conscientiousness is a personality trait dimension that globally describes people's propensity to be reliable, dutiful, orderly, ambitious, and cautious (Jackson et al. 2010). We chose this personality trait because it has high practical relevance due to its ability to predict important life outcomes and behaviors (Ozer and Benet-Martínez 2006). Here, we (1) fit a random forest model to predict the personality dimension of conscientiousness, (2) compute the introduced methods for grouped feature importance (GOPFI, GPMFI, GSI, LOGI, LOGO), (3) use the proposed sequential grouped feature importance procedure to investigate which groups are most important in combination, and (4) visualize the effect of different groups with CFEPs. Once the importance of individual groups has been quantified, CFEPs can be helpful to further explore the variables in these groups with regard to the criterion variable of interest (i.e., conscientiousness) to generate new hypotheses for future research.

In Fig. 10, we show a sequential procedure for our personality prediction example. The figure shows that the groups *overall phone usage* and *app usage* lead to the best model performance if used alone and, in many cases, lead to even better performances if combined. The results also suggests that if only one group can be selected, the initial selection of the feature group app usage more often leads to the smallest expected loss (mean MSE = 0.519). For a practical application, this would indicate that if only one type of feature may be collected from smartphones to predict the personality trait conscientiousness, features on app usage should be used. If two groups of data can

¹¹ <https://osf.io/kqjhr/>.

Table 5 Grouped feature importance values for predicting the personality trait conscientiousness based on MSE

Group	GOPFI	GPFI	GSI	LOGI	LOGO
Mobility (Mo)	-0.002 (± 0.011)	-0.002 (± 0.001)	0.000 (± 0.003)	-0.011 (± 0.075)	0.000 (± 0.006)
Music (Mu)	-0.001 (± 0.011)	0.002 (± 0.002)	0.001 (± 0.006)	-0.019 (± 0.074)	0.001 (± 0.012)
Communication and social (C)	0.000 (± 0.008)	0.001 (± 0.003)	0.004 (± 0.006)	0.008 (± 0.070)	0.001 (± 0.010)
Overall phone usage (O)	0.007 (± 0.011)	0.009 (± 0.003)	0.012 (± 0.008)	0.032 (± 0.080)	0.009 (± 0.014)
App usage (A)	0.032 (± 0.009)	0.028 (± 0.005)	0.031 (± 0.012)	0.041 (± 0.069)	0.011 (± 0.019)

All values were calculated using a resampling method (10-times cross-validation)

be collected, overall phone usage should also be added (mean MSE = 0.513). Finally, the plot indicates that in some cases ($n = 9$), the additional consideration of music listening behaviors in the model could lead to additional, small improvements of the expected loss (mean MSE = 0.508). If a feature group is not added, this means that it did not make a significant contribution in this iteration of the data split. Interestingly, the feature group *music* alone shows very low (or even negative) grouped feature importance scores. This would mean that music features are only predictive in the presence of other features.

To additionally explore meaningful and predictive directions in the feature space of the app usage group, we use CFEPs for the visualization. Subplot (a) in Fig. 11 shows that combinations of higher values in features on weather app usage on average lead to higher mean values in the personality trait conscientiousness. The increased frequency in weather app usage could signify the propensity of conscientious people to be prepared for future eventualities (e.g., bad weather; Jackson et al. 2010). Subplot (b) shows an interesting non-monotonic relationship between the number of different apps used each day and the mean value in conscientiousness. Subplot (c) shows that the combinations of higher values in overall phone activities lead to lower mean values in conscientiousness. Finally, plot (d) shows a similar, negative effect pattern with regard to music listening behaviors.

8 Conclusion

We introduced various techniques to analyze the importance and effect of user-defined feature groups on predictions of ML models. We provided formal definitions and dis-

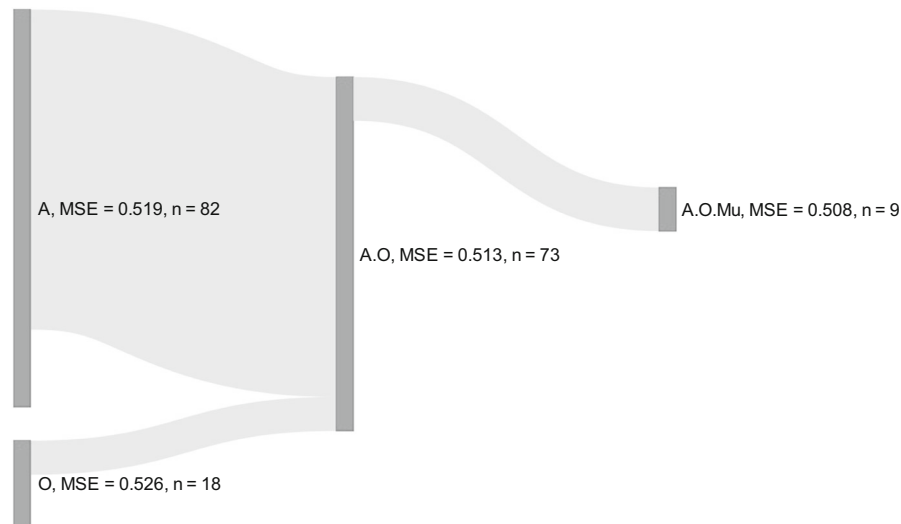


Fig. 10 Sequential grouped feature importance procedure for smartphone sensor data predicting *conscientiousness*. 100 times repeated subsampling. Inner resampling strategy: 10-fold cross-validation. Improvement threshold $\delta = 0.01$. Abbreviations: app-usage (A), communication & social (C), music (Mu), overall phone activity (O), mobility (Mo). Vertical bars show one step in the greedy forward search algorithm. Height of the vertical bars represent the number of subsampling iterations in which a combination of groups was chosen (for example, out of 100 subsampling iterations the group app-usage (A) was chosen 82 times as the best first group). Streams indicate the proportion of iterations that additionally benefited from a consequent step. Only streams containing at least 5 iterations and better mean performance at the end are displayed

tion criteria for grouped feature importance methods and distinguished between permutation- and refitting-based methods. For both approaches, we defined two calculation strategies that either start with a null model or with the full model. Based on these two definitions, we introduced Shapley importance scores for groups, which we defined for permutation as well as refitting methods. Moreover, we introduced as our first main contribution a sequential grouped feature importance procedure to find good and stable combinations of feature groups. To contrast the newly proposed methods with existing ones, we compared them for different scenarios. The key recommendations for the user can be summarized for four scenarios: (1) If high correlations between groups are present, refitting methods should be preferred over permutation methods, since they often deliver more meaningful results in these scenarios. Moreover, if the number of groups is reasonably small, refitting methods become computationally feasible. (2) If a sparse set of feature groups is of interest (e.g., due to data availability), the introduced sequential procedure can be useful. It provides insights regarding the most important groups: which sparse group combinations are stable in the sense that they are frequently selected and achieve a good performance. These criteria can be critically informative in situations where feature groups must be obtained from different data sources that are associated with further costs. (3) If the correlation strengths of features within groups are very diverse, all of the introduced methods might fail to reflect the true underlying importance of the feature groups. The size of this effect

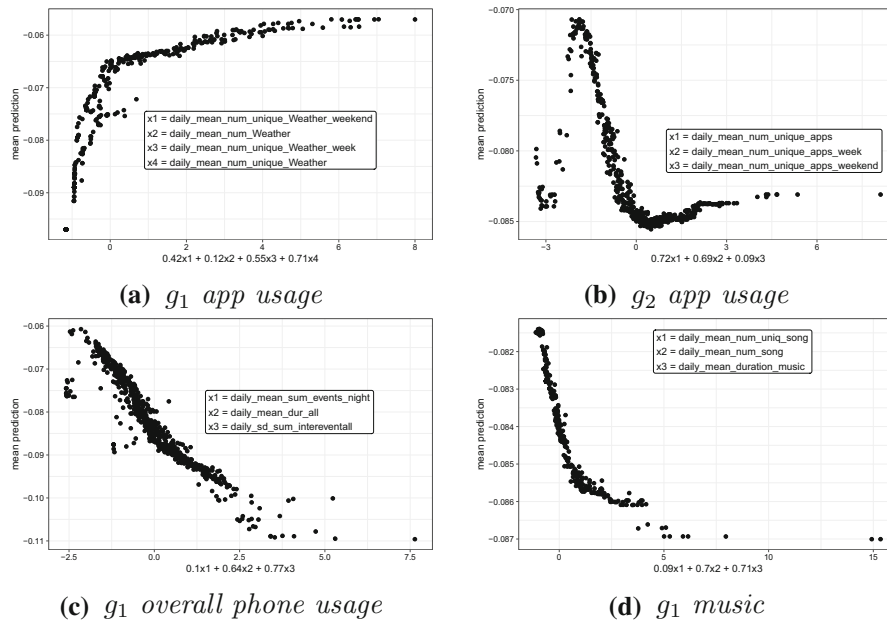


Fig. 11 CFEPs for the prediction of the personality trait conscientiousness. g_1 describes the first principal component of the respective group, and g_2 describes the second. More details about the features can be found in Appendix D and on the supplemental website https://compstat-lmu.shinyapps.io/Personality_Prediction/ for Stachl et al. (2020a)

depends heavily on how well the fitted model captures the true underlying relationship between features. Especially when using random forests, we showed that all of the methods lead to misleading results. (4) Groups with many features might tend to have a higher grouped importance score than groups with fewer features. Normalizing the grouped importance score leads to an average score per feature. However, this might result in choosing groups where grouped scores are smaller than those of other groups and, hence, contain less (performance-based) information than others. When using GSI, users can extract additional feature-level information to gain more insights into the group scores. Specifically, we showed that single feature Shapley importance scores add up to GSI when no higher-order interactions between groups are present. As third main contribution we proposed the CFEP, which is another global interpretation method that allows visualizations of the combined effect of multiple features on the prediction of an ML model. By applying a sparse SPCA, we received more meaningful and interpretable results for the final CFEPs compared to its unsupervised counterpart. We also demonstrated the suitability of the method in our real data example from computational psychology. Although, we only considered a numeric feature space here, all methods are in general also applicable to mixed feature spaces. However, in the presence of categorical features, a suitable dimension reduction method for CFEP must be chosen.

Here, we have focused on knowledge-driven feature groupings. However, the introduced methods could also be applied to data-driven groups (e.g., via shared variance).

Notably, their interpretation is only meaningful if groups can be described by some underlying factor. This might be a good application for interpretable latent variables to find causal relationships between feature groups and predictions of ML models. Additionally, with regard to highly correlated feature groups that cannot be grouped naturally, a data-driven approach might be more suitable.

It is our goal that this article not only provides a helpful reference for researchers in selecting appropriate interpretation methods when features can be grouped, but also that it inspires future research in this area.

Author Contributions Conceptualization: QA, JH, GC, BB; Methodology: QA, JH, GC; Formal analysis and investigation: QA, JH, GC; Writing - original draft preparation: QA, JH; Writing - review and editing: GC, CS, BB; Investigation: QA, JH; Visualization: QA, JH; Validation: QA, JH, GC; Software: QA, JH; Funding acquisition: GC, CS, BB; Supervision: GC, BB.

Funding Open Access funding enabled and organized by Projekt DEAL. This work has been partially supported by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A, the Bavarian State Ministry of Science and the Arts in the framework of the Centre Digitisation.Bavaria (ZD.B), the Bavarian Ministry of Economic Affairs, Regional Development and Energy as part of the program “Bayerischen Verbundförderprogramms (BayVFP)—Förderlinie Digitalisierung—Förderbereich Informations- und Kommunikationstechnik” under the Grant DIK-2106-0007 // DIK0260/02, a Google research grant, the LMU-excellence initiative, and the National Science Foundation (NSF) Award SES-1758835. The authors of this work take full responsibility for its content.

Availability of data and materials All data are created or provided in the following public git-repository: https://github.com/slds-lmu/grouped_feat_imp_and_effects.

Code Availability The implementation of the proposed methods and reproducible scripts for the experimental analysis are provided in the following public git-repository: https://github.com/slds-lmu/grouped_feat_imp_and_effects.

Declarations

Conflict of interest Not applicable.

Consent to participate Not applicable.

Ethics approval Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendix A Motivational example for grouped importance methods

In some settings, permuting single features individually might not be meaningful, for example, when categorical features are dummy-encoded. Table 6 shows for two

Table 6 We draw 1000 samples of two independent categorical random variables $X_1, X_2 \in \{1, 2, 3, 4\}$ where the categories 1 and 2 occur four times more frequently than 3 and 4

Method	X_1	$X_{2,2}, X_{2,3}, X_{2,4}$	$X_{2,2}$	$X_{2,3}$	$X_{2,4}$
Individually permuted	2.63	–	2.45	1.00	1.71
Group-wise permuted	2.63	2.65	–	–	–

Consider the target $y = 5 \cdot \mathbb{1}_{X_1 \neq 1} + 5 \cdot \mathbb{1}_{X_2 \neq 1} + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 1)$. Both categorical features have the same influence on the target. We explicitly dummy encode X_2 using $X_2 = 1$ as the reference category to obtain 3 binary features $X_{2,k} = \mathbb{1}_{X_2=k}, k \in \{2, 3, 4\}$. We fit a linear model using the categorical feature X_1 and the binary features $X_{2,2}, X_{2,3}, X_{2,4}$. Here, we want to illustrate why it makes more sense to permute the 3 binary features jointly rather than individually, since they naturally belong together. As expected, permuting the binary features $X_{2,2}, X_{2,3}, X_{2,4}$ jointly as a group yields a comparable importance to X_1 . However, permuting each binary feature individually gives different importance scores making it unclear how important X_2 is compared to X_1

equally important categorical features that if one feature is dummy-encoded (here: X_2), then all resulting binary features must be permuted as a group to obtain a comparable importance score to X_1 . Hence, settings like in Table 6 or as described in Sects. 1 or 1.2 point out the need of grouped importance methods.

Appendix B Shapley importance

B.1 Properties of the grouped Shapley importance

For single features¹² $x_i \in \{1, \dots, p\}$, which are divided into l groups, we define the marginal contribution for x_i as

$$\Delta_{\{x_i\}}(S) = v(S \cup \{x_i\}) - v(S),$$

for $S \subset \{1, \dots, p\} \setminus \{x_i\}$. The Shapley importance for single features $\phi(x_i)$ can also be defined analogously to (15). One interesting question is, does the GSI for a group $G \subset \{1, \dots, p\}$ decompose into the sum of Shapley importances of features in G ? In the following, we want to analyze the remainder

$$R = \phi(G) - \sum_{i \in G} \phi(x_i). \tag{B1}$$

Similar to the functional ANOVA decomposition (Hooker 2004), we assume, that the value function for a coalition $S \subset \{1, \dots, p\}$ can be broken down into main and interaction effects

$$v(S) = v_0 + \sum_{x_i \in S} v(x_i) + \sum_{i \neq j} \epsilon_{ij} + \sum_{i \neq j \neq k} \epsilon_{ijk} + \dots, \tag{B2}$$

¹² Remember the one-to-one association of the numbers $1, \dots, p$ and the features $\mathbf{x}_1, \dots, \mathbf{x}_p$

where $\epsilon_{i\dots m}$ is the effect of the interaction between the features $x_i, \dots, x_m \in S$. A needed requirement to apply this decomposition is that each of the functional terms has zero means, hence they need to be centralized. The considered intercept shift is stored in v_0 . To receive a unique decomposition, the orthogonality between the functional terms needs to be fulfilled which is not the case in the presence of correlated features. Hooker (2007) therefore suggests the generalized functional ANOVA which replaces the orthogonality property with a hierarchical orthogonality condition and which is a weighted version of the standard functional ANOVA (Hooker 2004). However, we do not try to estimate or calculate the decomposed function terms, we only use the (valid) assumption that a function can be decomposed as in Eq. (B2) to show how GSI relates to Shapley importance for individual features. Hence, we are not directly interested in a unique solution of the decomposition.

With the assumption in Eq. (B2), it follows that the Shapley importance of a single feature x_1 (without loss of generality) can be written as

$$\phi(x_1) = v(x_1) + \frac{1}{2} \left(\sum_{i \neq 1}^p \epsilon_{1i} \right) + \frac{1}{3} \left(\sum_{i \neq j \neq 1}^p \epsilon_{1ij} \right) + \dots + \frac{1}{p} \epsilon_{1\dots p}. \quad (\text{B3})$$

The value function of the feature x_1 contributes to the Shapley importance with the weight 1 and all possible interaction effects with feature x_1 contribute with the reciprocal length of the interaction effect. We proved this assertion in Appendix B.2. Similar to (B3), the GSI of a group G_1 (w.l.o.g.) can be written as

$$\phi(G_1) = v(G_1) + \frac{1}{2} \left(\sum_{i \neq 1}^k \epsilon_{G_1 G_i} \right) + \frac{1}{3} \left(\sum_{i \neq j \neq 1}^k \epsilon_{G_1 G_i G_j} \right) + \dots + \frac{1}{k} \epsilon_{G_1 \dots G_k} \quad (\text{B4})$$

where $\epsilon_{G_1 \dots G_k}$ is the (non-computable) interaction effect between features of groups G_1, \dots, G_k , where each group provides at least one feature. By using Eq. (B2) on $v(G_1)$, we get:

$$v(G_1) = \sum_{i \in G_1} v(x_i) + \sum_{i \neq j \in G_1} \epsilon_{ij} + \sum_{i \neq j \neq k \in G_1} \epsilon_{ijk} + \dots \quad (\text{B5})$$

Looking back at Eq. (B1), a lot of terms cancel out by using Eqs. (B3) and (B5). The term $v(G_1)$, meaning all main effects $v(x_i), i \in G_1$, and all interaction effects $\epsilon_{i, \dots, k}, 1 \leq k \leq |G_1|$ between features within G_1 , cancels out entirely.¹³ Furthermore, at least all two-way interaction effects between groups $\epsilon_{G_1 G_i}, i = 2, \dots, k$ cancel out. A combination of higher-order interaction terms between features of G_1 and $\{1, \dots, p\} \setminus G_1$ remain.¹⁴ This means that the remainder R is (usually) not equal to zero in case the applied algorithm learned a higher-order interaction between features

¹³ Note, $v(G_1)$ cancels out, meaning that these interaction terms cannot be computed directly but are assumed to affect the “payout” of the value function.

¹⁴ They mostly only partly cancel out, depending on the number of features within the groups G_1, \dots, G_k .

of the regarded group and other groups. The higher the remainder, the larger the higher-order interaction effect. Thus, the remainder can be used as a quantification of learned higher-order interaction effects between features of different groups.

B.2 Proof of Properties

Assume, that the value function for a coalition $S \subset \{x_1, \dots, x_p\}$ can be broken down into main and interaction effects:

$$v(S) = \sum_{x_i \in S} v(x_i) + \sum_{i_1 \neq i_2} \epsilon_{i_1 i_2} + \sum_{i_1 \neq i_2 \neq i_3} \epsilon_{i_1 i_2 i_3} + \dots,$$

the Shapley importance of a single feature x_1 can be written as

$$\phi(x_1) = v(x_1) + \frac{1}{2} \left(\sum_{i \neq 1}^p \epsilon_{1i} \right) + \frac{1}{3} \left(\sum_{i \neq j \neq 1}^p \epsilon_{1ij} \right) + \dots + \frac{1}{p} \epsilon_{1\dots p}.$$

Proof Let $N = \{x_2, \dots, x_p\}$. The general formula for the Shapley importance is given by:

$$\phi_p(x_1) = \sum_{S \subset N \setminus \{x_1\}} \frac{(p-1-|S|)! \cdot |S|!}{p!} (v(S \cup \{x_1\}) - v(S)) \tag{B6}$$

With assumption (B2) the term $v(S \cup \{x_1\}) - v(S)$ will reduce to:

$$v(S \cup \{x_1\}) - v(S) = v(x_1) + \sum_{i_1 \neq 1}^p \epsilon_{1i_1} + \dots + \sum_{i_1 \neq \dots \neq i_{|S|} \neq 1}^p \epsilon_{1i_1 \dots i_{|S|}} \tag{B7}$$

It is the sum of $v(x_1)$ and all interactions with feature x_1 of sizes $2, \dots, |S| + 1$. All other terms without feature x_1 cancel out.

Equation (B6) consists of many summands of the form (B7). The term $v(x_1)$ appears once for every subset $S \subset N \setminus \{x_1\}$. There are $\binom{p-1}{|S|}$ different subsets of size $|S|$. Only looking at the summands with the term $v(x_1)$, Eq. (B6) reduces to

$$\sum_{|S|=0}^{p-1} \frac{(p-1-|S|)! \cdot |S|!}{p!} \binom{p-1}{|S|} v(x_1) = v(x_1). \tag{B8}$$

For the interaction terms, we first start counting the interaction term ϵ_{12} of size 2, as an example. For $|S| = 0$, there are zero terms of ϵ_{12} . For $|S| = 1$, the term ϵ_{12} only appears once, when $S = \{x_2\}$. For $|S| = 2$, the term ϵ_{12} appears $p - 2$ times, once for each subset $S = \{x_2, x_j\}$, for $3 \leq j \leq p$. For $|S| = 3$, we have $\binom{p-2}{2}$ times the term ϵ_{12} , again, once for each subset $S = \{x_2, x_j, x_k\}$, for $3 \leq j \neq k \leq p$. This pattern goes on until there are $\binom{p-2}{p-2}$ terms of ϵ_{12} for $|S| = p - 1$. Now, we look at

the interaction terms $\epsilon_{1i_1 \dots i_{k-1}}$ of size k . Following the pattern, which we just derived, there are zero terms of $\epsilon_{1i_1 \dots i_{k-1}}$ for $|S| \leq k - 2$ and $\binom{p-k}{|S|-k+1}$ terms of $\epsilon_{1i_1 \dots i_{k-1}}$ for $k \leq |S| \leq p - 1$. If we only look at the interaction terms $\epsilon_{1i_1 \dots i_{k-1}}$ of size k and following the Eq. (B6), we get

$$\sum_{|S|=k-1}^{p-1} \frac{(p-1-|S|)! \cdot |S|!}{p!} \binom{p-k}{|S|-k+1} \epsilon_{1i_1 \dots i_{k-1}} = \frac{1}{k} \epsilon_{1i_1 \dots i_{k-1}},$$

which was left to show the assertion. \square

Appendix C More details on dimension reduction techniques

C.1 Principal component analysis

PCA only considers the data matrix \mathbf{X} and does not take the target vector \mathbf{Y} into account. This procedure is thus unsupervised.

Given a centering Matrix

$$\mathbf{H} = \mathbf{I} - n^{-1} \mathbf{e} \mathbf{e}^T, \quad (\text{C9})$$

where \mathbf{e} is an n -dimensional vector of all ones. The centered matrix is $\mathbf{X}_C = \mathbf{H}\mathbf{X}$. The sample covariance matrix of \mathbf{X} can be written as:

$$\mathbf{S}_X := \frac{1}{n} \mathbf{X}_C^T \mathbf{X}_C = \frac{1}{n} \mathbf{X}^T \mathbf{H} \mathbf{H} \mathbf{X} \quad (\text{C10})$$

The goal is to maximize the total variance of projected data, which is equivalent to maximizing trace of the sample covariance matrix. Equation (C10) can also be written as $\mathbf{S}_X = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_C^{(i)} \mathbf{x}_C^{(i)T}$, where $\mathbf{x}_C^{(i)}$ corresponds to the i -th row of \mathbf{X}_C . By projecting each data point by some unknown vectors \mathbf{v}_j , $j = 1, \dots, p$, we get the projected variance for each $j = 1, \dots, p$, which is:

$$\frac{1}{n} \sum_{i=1}^n \mathbf{v}_j^T \mathbf{x}_C^{(i)} \mathbf{x}_C^{(i)T} \mathbf{v}_j = \mathbf{v}_j^T \left(\frac{1}{n} \sum_{i=1}^n \mathbf{x}_C^{(i)} \mathbf{x}_C^{(i)T} \right) \mathbf{v}_j = \mathbf{v}_j^T \mathbf{S}_X \mathbf{v}_j.$$

Let $\mathbf{V} \in \mathbb{R}^{p \times p}$ be the full projection matrix. The projected total variance is $\text{tr}(\mathbf{V}^T \mathbf{S}_X \mathbf{V})$, and by ignoring constant terms, PCA finds a solution to the problem

$$\underset{\mathbf{V}}{\text{argmax}} \text{tr}(\mathbf{V}^T \mathbf{S}_X \mathbf{V}) = \underset{\mathbf{V}}{\text{argmax}} \text{tr}(\mathbf{V}^T \mathbf{X}^T \mathbf{H} \mathbf{H} \mathbf{X} \mathbf{V}) \quad (\text{C11})$$

with an Eigen decomposition of the covariance matrix \mathbf{S}_X . The resulting Eigen vectors thus maximize the variation of projected data.

C.2 Measuring statistical dependence with Hilbert Schmidt norms

In Gretton et al. (2005) a more generalized measure of dependence between variables X and Y was introduced:

Two random variables X and Y are independent if and only if any bounded continuous function of them are uncorrelated.

In more detail, this means that any pairs (X, Y) , (X, Y^2) , (X^2, Y) , $(\cos(X), \log(Y))$, ... have to be uncorrelated. The resulting independence measure is called the Hilbert-Schmidt Independence Criterion (HSIC). For the analysis of this independence measure, it is necessary to analyze functions on random variables. Therefore theory of Hilbert spaces and concepts of functional analysis are necessary for a thorough analysis, but they are not part of this paper. For an extensive discussion of Hilbert spaces, especially reproducing kernel hilbert spaces (RKHS) we refer to Hein and Bousquet (2004).

Let \mathcal{F} be a separable RKHS containing all bounded continuous functions from \mathcal{X} to \mathbb{R} . The associated kernel shall be denoted by $\mathbf{K} \in \mathbb{R}^{n \times n}$, with $\mathbf{K}_{ij} = k(x_i, x_j)$. Concurrently, let \mathcal{G} be a separable RKHS with bounded continuous functions from \mathcal{Y} to \mathbb{R} and associated kernel $\mathbf{L} \in \mathbb{R}^{n \times n}$, with $\mathbf{L}_{ij} = l(y_i, y_j)$.

We are particularly interested in the cross variance between f and g :

$$\text{Cov}(f(x), g(y)) = \mathbb{E}_{x,y}[f(x)g(y)] - \mathbb{E}_x[f(x)]\mathbb{E}_y[g(y)] \quad (\text{C12})$$

A function, which maps one element from one hilbert space to another hilbert space is called *operator*. A theorem (see e.g. Fukumizu et al. 2004) states, that there exists a unique operator $C_{X,Y} : \mathcal{G} \rightarrow \mathcal{F}$ with

$$\langle f, C_{x,y}(g) \rangle_{\mathcal{F}} = \text{Cov}(f(x), g(y)). \quad (\text{C13})$$

The Hilbert-Schmidt Independence Criterion (HSIC) is defined as the squared Hilbert-Schmidt norm of the cross-covariance operator C :

$$\text{HSIC}(P_{\mathcal{X},\mathcal{Y}}, \mathcal{F}, \mathcal{G}) = \|C_{x,y}\|_{HS}^2 \quad (\text{C14})$$

$\|C_{x,y}\|_{HS}^2 = 0$ if and only if the random variables \mathcal{X} and \mathcal{Y} are independent. For a detailed discussion and derivation of the HSIC independence measure, we refer to Gretton et al. (2005). The HSIC measure was used for feature selection in Song et al. (2007) or for supervised principal components in Barshan et al. (2011).

C.2.1 Empirical HSIC

For a dataset $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$ the empirical HSIC is:

$$\text{HSIC}(\mathcal{D}, F, G) = (n-1)^{-2} \text{tr}(\mathbf{KHLH}) = (n-1)^{-2} \text{tr}(\mathbf{HKHL}), \quad (\text{C15})$$

where \mathbf{H} is the centering matrix from (C9). A high level of dependency between two kernels yields a high HSIC value.

C.3 Supervised sparse principal components

In the process of finding interpretable latent variables, which also incorporate dependencies to a target variable, the Sparse Supervised Principal Components (SPCA), which was introduced in Sharifzadeh et al. (2017), is a suitable method for our application.

For sparse SPCA the kernel matrix K is defined as $K = X V V^T X^T$ with a constraint for unit length and an L_1 penalty for sparsity. By ignoring constant terms, we get the optimization problem:

$$\operatorname{argmax}_{\mathbf{V}} \operatorname{tr}(\mathbf{H}\mathbf{K}\mathbf{H}\mathbf{L}) = \operatorname{argmax}_{\mathbf{V}} \operatorname{tr}(\mathbf{H}\mathbf{X}\mathbf{V}\mathbf{V}^T\mathbf{X}^T\mathbf{H}\mathbf{L}) \quad (\text{C16})$$

$$= \operatorname{argmax}_{\mathbf{V}} \operatorname{tr}(\mathbf{V}^T\mathbf{X}^T\mathbf{H}\mathbf{L}\mathbf{H}\mathbf{X}\mathbf{V}) \quad (\text{C17})$$

$$s.t. \quad \mathbf{V}^T\mathbf{V} = \mathbf{I}, \quad |\mathbf{V}| \leq c. \quad (\text{C18})$$

Note, that without the sparsity constraint, (C17) reduces to (C11), when choosing $\mathbf{L} = \mathbf{I}$. Already explained in Barshan et al. (2011), PCA is a special form of their Supervised PCA, where setting $\mathbf{L} = \mathbf{I}$ is a kernel, which only captures similarity between a point and itself. Maximizing dependency between \mathbf{K} and the identity matrix corresponds to retaining maximal diversity between observations.

Now, an arbitrary \mathbf{L} can be decomposed as $\mathbf{L} = \Delta\Delta^T$, since \mathbf{L} , as a kernel matrix, is positive definite and symmetric. Defining $\Psi := \Delta^T\mathbf{H}\mathbf{X} \in \mathbb{R}^{n \times p}$, the objective function (C17) can be rewritten as:

$$\operatorname{argmax}_{\mathbf{V}} \operatorname{tr}(\mathbf{V}^T\Psi^T\Psi\mathbf{V}) \quad s.t. \quad \mathbf{V}^T\mathbf{V} = \mathbf{I}, \quad |\mathbf{V}| \leq c. \quad (\text{C19})$$

Using the singular value decomposition (SVD), the matrix Ψ with $\operatorname{rank}(\Psi) = m \leq n$ can be written as a product of matrices:

$$\Psi = \mathbf{U}\Lambda\mathbf{V}^T \quad s.t. \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}_n, \quad \mathbf{V}\mathbf{V}^T = \mathbf{I}_p, \quad \Lambda = I(\lambda_1, \dots, \lambda_m, 0, \dots, 0), \quad (\text{C20})$$

where $\mathbf{U} \in \mathbb{R}^{n \times n}$ and $\mathbf{V} \in \mathbb{R}^{p \times p}$ are orthogonal matrices, and $\Lambda \in \mathbb{R}^{n \times p}$ is a diagonal matrix, with descending diagonal entries $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m \geq 0$. It is easy to see that the columns of \mathbf{V} are Eigen vectors of the matrix $\Psi^T\Psi$, since the following Eigen value decomposition holds:

$$\Psi^T\Psi = \mathbf{V}\Lambda\mathbf{U}^T\mathbf{U}\Lambda\mathbf{V}^T = \mathbf{V}(\Lambda^2)\mathbf{V}^T. \quad (\text{C21})$$

The sparse SPCA problem (C19) now becomes a matrix decomposition problem of the matrix Ψ , when adding an L_1 penalty on the matrix \mathbf{V} , since the columns of \mathbf{V} , being Eigen vectors of $\Psi^T\Psi$, maximize $\operatorname{tr}(\mathbf{V}^T\Psi^T\Psi\mathbf{V})$.

With an L_1 penalty on \mathbf{V} , this problem is a *penalized matrix decomposition* problem (PMD, Witten et al. (2009)).

Recalling our original problem of finding interpretable latent variables that also depend on a target variable, the rank m matrix decomposition of Ψ may not be desirable. It can be shown (e.g. Eckart and Young 1936) that the best low rank ($r \leq m$) approximation of Ψ is calculated by the first r singular values of Λ and the first r singular vectors of \mathbf{U} and \mathbf{V} . With \mathbf{u}_i being the i -th column of \mathbf{U} and \mathbf{v}_i being the i -th column of \mathbf{V} , the best low rank approximation can thus be written as:

$$\sum_{i=1}^r \lambda_i \mathbf{u}_i \mathbf{v}_i^T = \underset{\hat{\Psi}}{\operatorname{argmin}} \|\Psi - \hat{\Psi}\|_F^2, \tag{C22}$$

subject to the squared Frobenius-norm ($A \in \mathbb{R}^{m \times n}$: $\|A\|_F^2 = \sum_{i=1}^n \sum_{j=1}^m |a_{ij}|^2$). The following equality was demonstrated in Witten et al. (2009):

$$\frac{1}{2} \|\Psi - \mathbf{U} \Lambda \mathbf{V}^T\|_F^2 = \frac{1}{2} \|\Psi\|_F^2 - \sum_{i=1}^r \mathbf{u}_i^T \Psi \mathbf{v}_i \lambda_i + \frac{1}{2} \sum_{i=1}^r \lambda_i^2. \tag{C23}$$

The minimization problem (C22) thus becomes a maximization problem, by ignoring the constant terms. Sharifzadeh et al. (2017) added additional L_2 constraints on \mathbf{u}_i and \mathbf{v}_i , an L_1 constraint on v_i for sparsity and an orthogonality constraint for u_i :

$$\underset{\mathbf{u}_i, \mathbf{v}_i}{\operatorname{argmax}} \mathbf{u}_i^T \Psi \mathbf{v}_i \text{ s.t. } \|\mathbf{u}_i\|_2 \leq 1, \|\mathbf{v}_i\|_2 \leq 1, \|\mathbf{v}_i\|_1 \leq c, \mathbf{u}_i \perp \mathbf{u}_1, \dots, \mathbf{u}_{i-1} \tag{C24}$$

The L_2 constraints do not force unit length to avoid non convex optimization problems. Witten et al. (2009) discuss how to solve many penalized matrix decomposition problems of this kind. Without the orthogonality constraint, they call this particular problem PMD(\cdot, L_1). The solution to this problem is discussed in detail in Sharifzadeh et al. (2017). A software implementation is available with the R-package PMA by Witten and Tibshirani (2020), which we will use for our demonstrations. Problem (C24) does not yield orthogonal sparse vectors \mathbf{v}_i , Witten et al. (2009) state that these vectors are unlikely to be very correlated, since the vectors \mathbf{v}_i are associated with orthogonal vectors $\mathbf{u}_i, i = 1, \dots, r$.

C.3.1 Choice of the Kernel

For sparse SPCA the kernel \mathbf{K} has been predefined as. The choice of the kernel \mathbf{L} , however, has a decisive impact on how the dependencies are modeled. Song et al. (2012) discuss the kernel choice for different situations. For binary classification, one may simply choose

$$l(y_i, y_j) = y_i y_j, \text{ where } y_i, y_j \in \{\pm 1\}, \tag{C25}$$

or a weighted version, giving different weights on positive and negative labels. For multiclass classification a possible kernel is

$$l(y_i, y_j) = c_y \delta_{y_i, y_j}, \text{ where } c_y > 0. \tag{C26}$$

For regression one can also use a linear kernel $l(y_i, y_j) = y_i, y_j$, but then only simple linear correlations between features and the target variable can be detected. A more universal choice is the radial basis function (RBF) kernel:

$$l(y_i, y_j) = \exp\left(-\frac{\|y_i - y_j\|^2}{2\sigma^2}\right). \quad (\text{C27})$$

The choice of the bandwidth $2\sigma^2$ is extremely important. For example, if $2\sigma^2 \rightarrow 0$, the matrix L becomes the identity matrix. Or if $2\sigma^2 \rightarrow \infty$, all entries of L are 1. In both cases, all relevant information of the dependency between features and the target variable is lost. Besides the bandwidth 2σ , the kernel matrix L depends only on the pairwise distances $\|y_i - y_j\|^2$. A reasonable, and heuristically well performing (Pfister et al. 2017) choice is $2\sigma^2 = \text{median}(\|y_i - y_j\|^2 : i > j)$. However, it might also be possible and advantageous to use other kernels that are selected to be particularly efficient in detecting certain kinds of dependencies.

C.3.2 Choice of c

Witten et al. (2009) explained how PMD can be used to impute missing data. The main idea is simply to exclude missing entries from the maximization problem (C24) and impute missing values by the low rank approximation matrix $U\Lambda V^T$. This procedure can also be used for finding optimal values for c by a cross-validation approach. The test data consists of leaving out some entries of the matrix Ψ (not entire rows or columns, but individual elements of the matrix), yielding a matrix with missing entries $\tilde{\Psi}$. For candidate values $c_i, i = 1, \dots, k$, calculate the $\text{PMD}(\cdot, L_1)$ and record the mean squared error over the missing elements of $\tilde{\Psi}$ and the estimate $U\Lambda V^T$. The true values of the missing values of $\tilde{\Psi}$ are available in the original data Ψ . The optimal value c^* corresponds to the best candidate value c_j , which minimizes the mean squared error.

However, such a cross-validation approach for the search for c is not always necessary. If the method is used as a descriptive method to better understand the underlying structure of the data, a small value of c can be chosen to achieve a desired sparsity.

Appendix D Feature description for smartphone sensor data

See Table 7.

Table 7 Description of features used for CFEPs in Sect. 7

Feature	Description
daily_mean_num_unique_Weather_weekend	Mean number of different weather apps used each day on weekends
daily_mean_num_Weather	Mean number of weather apps used each day
daily_mean_num_unique_Weather_week	Mean number of different weather apps used each day on weekdays
daily_mean_num_unique_Weather	Mean number of different weather apps used each day
daily_mean_num_unique_apps	Mean number of different apps used each day
daily_mean_num_unique_apps_week	Mean number of different apps used each day on weekdays
daily_mean_num_unique_apps_weekend	Mean number of different apps used each day on weekends
daily_mean_sum_events_night	Number of all events during the night averaged for each day
daily_mean_dur_all	Duration of all events averaged for each day
daily_sd_sum_intereventall	Sd of the sum of all inter-event time intervals for each day
daily_mean_num_uniq_song	Mean number of different songs listened to each day
daily_mean_num_song	Mean number of songs listened to each day
daily_mean_duration_music	Mean duration of music apps used each day

References

- Allaire J, Gandrud C, Russell K, et al (2017) networkD3: D3 JavaScript network graphs from R. <https://CRAN.R-project.org/package=networkD3>, R package version 0.4
- Alon U, Barkai N, Notterman DA et al (1999) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc Natl Acad Sci* 96(12):6745–6750
- Amoukou SI, Brunel NJB, Salaün T (2021) The shapley value of coalition of variables provides better explanations. [arXiv:2103.13342](https://arxiv.org/abs/2103.13342)
- Apley DW, Zhu J (2019) Visualizing the effects of predictor variables in black box supervised learning models. [arXiv:1612.08468](https://arxiv.org/abs/1612.08468)
- Bair E, Hastie T, Paul D et al (2006) Prediction by supervised principal components. *J Am Stat Assoc* 101(473):119–137
- Barshan E, Ghodsi A, Azimifar Z et al (2011) Supervised principal component analysis: visualization, classification and regression on subspaces and submanifolds. *Pattern Recogn* 44(7):1357–1371
- Berk R, Sherman L, Barnes G et al (2009) Forecasting murder within a population of probationers and parolees: a high stakes application of statistical learning. *J R Stat Soc A Stat Soc* 172(1):191–211
- Breiman L (2001) Random forests. *Mach Learn* 45(1):5–32
- Brenning A (2021) Transforming feature space to interpret machine learning models. [arXiv:2104.04295](https://arxiv.org/abs/2104.04295)
- Caputo B, Sim K, Furesjö F, et al (2002) Appearance-based object recognition using SVMS: Which kernel should I use. In: *Proceedings of the NIPS workshop on statistical methods for computational experiments in visual processing and computer vision*, Red Hook, NY, USA
- Casalichio G, Molnar C, Bischl B (2019) Visualizing the feature importance for black box models. Springer International Publishing. *Machine Learning and Knowledge Discovery in Databases*, pp 655–670
- Chakraborty D, Pal NR (2008) Selecting useful groups of features in a connectionist framework. *IEEE Trans Neural Netw* 19(3):381–396

- Cohen SB, Ruppin E, Dror G (2005) Feature selection based on the Shapley value. In: Kaelbling LP, Saffiotti A (eds) IJCAI-05, Proceedings of the nineteenth international joint conference on artificial intelligence, Edinburgh, Scotland, UK, July 30–August 5, 2005. Professional Book Center, pp 665–670
- Covert I, Lundberg SM, Lee SI (2020) Understanding global feature contributions with additive importance measures. *Adv Neural Inf Process Syst* 33:17212–17223
- de Mijolla D, Frye C, Kunesch M, et al (2020) Human-interpretable model explainability on high-dimensional data. *CoRR* [arXiv:2010.07384](https://arxiv.org/abs/2010.07384)
- Eckart C, Young G (1936) The approximation of one matrix by another of lower rank. *Psychometrika* 1(3):211–218
- Fisher A, Rudin C, Dominici F (2019) All models are wrong, but many are useful: learning a variable’s importance by studying an entire class of prediction models simultaneously. *J Mach Learn Res* 20(177):1–81
- Friedman JH (2001) Greedy function approximation: a gradient boosting machine. *Ann Stat*, 1189–1232
- Friedman J, Hastie T, Tibshirani R (2010) A note on the group lasso and a sparse group lasso. [arXiv:1001.0736](https://arxiv.org/abs/1001.0736)
- Fuchs K, Scheipl F, Greven S (2015) Penalized scalar-on-functions regression with interaction term. *Comput Stat Data Anal* 81:38–51
- Fukumizu K, Bach FR, Jordan MI (2004) Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *J Mach Learn Res* 5:73–99
- Goldberg LR (1990) An alternative “description of personality”: the big-five factor structure. *J Person Soc Psychol* 59:1216–1229
- Goldstein A, Kapelner A, Bleich J et al (2013) Peeking inside the black box: visualizing statistical learning with plots of individual conditional expectation. *J Comput Gr Stat* 24:44–65
- Gregorova M, Kalousis A, Marchand-Maillet S (2018) Structured nonlinear variable selection. In: Globerson A, Silva R (eds) Proceedings of the thirty-fourth conference on uncertainty in artificial intelligence, UAI 2018, Monterey, California, USA, August 6–10, 2018. AUAI Press, pp 23–32
- Gregorutti B, Michel B, Saint-Pierre P (2015) Grouped variable importance with random forests and application to multiple functional data analysis. *Comput Stat Data Anal* 90:15–35
- Gretton A, Bousquet O, Smola A, et al (2005) Measuring statistical dependence with Hilbert-Schmidt norms. In: International conference on algorithmic learning theory. Springer, pp 63–77
- Guyon I, Weston J, Barnhill S et al (2002) Gene selection for cancer classification using support vector machines. *Mach Learn* 46(1–3):389–422
- Harari GM, Gosling SD, Wang R et al (2015) Capturing situational information with smartphones and mobile sensing methods. *Eur J Pers* 29(5):509–511
- Harari GM, Lane ND, Wang R et al (2016) Using smartphones to collect behavioral data in psychological science: opportunities, practical considerations, and challenges. *Perspect Psychol Sci* 11(6):838–854
- Harari GM, Müller SR, Aung MS et al (2017) Smartphone sensing methods for studying behavior in everyday life. *Curr Opin Behav Sci* 18:83–90
- Harari GM, Müller SR, Stachl C et al (2019) Sensing sociability: individual differences in young adults’ conversation, calling, texting, and app use behaviors in daily life. *J Person Soc Psychol* 119:204
- He Z, Yu W (2010) Stable feature selection for biomarker discovery. *Comput Biol Chem* 34:215–225
- Hein M, Bousquet O (2004) Kernels, Associated structures and generalizations, Max Planck Institute for Biological Cybernetics
- Shipp MA, Ross KN, Tamayo P et al (2002) Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nat Med* 8(1):68–74
- Hooker G (2004) Discovering additive structure in black box functions. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 575–580
- Hooker G (2007) Generalized functional anova diagnostics for high-dimensional functions of dependent variables. *J Comput Graph Stat* 16(3):709–732
- Hooker G, Mentch L (2019) Please stop permuting features: an explanation and alternatives. [arXiv:1905.03151](https://arxiv.org/abs/1905.03151)
- Jackson JJ, Wood D, Bogg T et al (2010) What do conscientious people do? Development and validation of the behavioral indicators of conscientiousness (bic). *J Res Pers* 44(4):501–511
- Jaeger J, Sengupta R, Ruzzo W (2003) Improved gene selection for classification of microarrays. *Pac Symp Biocomput Pac Symp Biocomput* 8:53–64
- Jolliffe IT (1986) Principal component analysis. Springer, New York
- Kolenik T, Gams M (2021) Intelligent cognitive assistants for attitude and behavior change support in mental health: state-of-the-art technical review. *Electronics* 10(11):1250

- Lei J, G'Sell M, Rinaldo A et al (2018) Distribution-free predictive inference for regression. *J Am Stat Assoc* 113(523):1094–1111
- Lipton ZC (2018) The myths of model interpretability: in machine learning, the concept of interpretability is both important and slippery. *Queue* 16(3):31–57
- Lozano AC, Abe N, Liu Y et al (2009) Grouped graphical granger modeling for gene expression regulatory networks discovery. *Bioinformatics* 25(12):i110–i118
- Lundberg SM, Erion GG, Lee S (2018) Consistent individualized feature attribution for tree ensembles. *CoRR* [arXiv:1802.03888](https://arxiv.org/abs/1802.03888)
- Lundberg SM, Lee SI (2017) A unified approach to interpreting model predictions. In: Proceedings of the 31st international conference on neural information processing systems. Curran Associates Inc., Red Hook, NY, USA, NIPS'17, pp 4768–4777
- Meier L, Van De Geer S, Bühlmann P (2008) The group lasso for logistic regression. *J R Stat Soc Ser B (Stat Methodol)* 70(1):53–71
- Meinshausen N, Bühlmann P (2010) Stability selection. *J R Stat Soc Ser B (Stat Methodol)* 72(4):417–473
- Miller G (2012) The smartphone psychology manifesto. *Perspect Psychol Sci* 7(3):221–237
- Molnar C (2019) Interpretable machine learning. <https://christophm.github.io/interpretable-ml-book/>
- Molnar C, König G, Bischl B, et al (2020a) Model-agnostic feature importance and effects with dependent features—a conditional subgroup approach. [arXiv:2006.04628](https://arxiv.org/abs/2006.04628)
- Molnar C, König G, Herbringer J, et al (2020b) General pitfalls of model-agnostic interpretation methods for machine learning models. *arXiv preprint* [arXiv:2007.04131](https://arxiv.org/abs/2007.04131)
- Nicodemus K, Malley J, Strobl C, et al (2010) The behaviour of random forest permutation-based variable importance measures under predictor correlation. *BMC Bioinform* 11–110
- Onnela JP, Rauch SL (2016) Harnessing smartphone-based digital phenotyping to enhance behavioral and mental health. *Neuropsychopharmacology* 41(7):1691–1696
- Ozer DJ, Benet-Martínez V (2006) Personality and the prediction of consequential outcomes. *Annu Rev Psychol* 57:401–421
- Park MY, Hastie T, Tibshirani R (2006) Averaged gene expressions for regression. *Biostatistics* 8(2):212–227
- Pfister N, Bühlmann P, Schölkopf B et al (2017) Kernel-based tests for joint independence. *J R Stat Soc Ser B (Stat Methodol)* 80(1):5–31
- Rachuri KK, Musolesi M, Mascolo C, et al (2010) Emotionsense: a mobile phones based adaptive platform for experimental social psychology research. In: *UbiComp'10—Proceedings of the 2010 ACM conference on ubiquitous computing*
- Raento M, Oulasvirta A, Eagle N (2009) Smartphones: an emerging tool for social scientists. *Sociol Methods Res* 37(3):426–454
- Rapaport F, Barillot E, Vert JP (2008) Classification of Arraycgh data using fused SVM. *Bioinformatics* 24(13):i375–i382
- Saeb S, Lattie EG, Schueller SM et al (2016) The relationship between mobile phone location sensor data and depressive symptom severity. *PeerJ* 4:e2537
- Schoedel R, Au Q, Völkel ST et al (2018) Digital footprints of sensation seeking. *Zeitschrift für Psychologie* 226(4):232–245
- Schoedel R, Pargent F, Au Q et al (2020) To challenge the morning lark and the night owl: using smartphone sensing data to investigate day-night behaviour patterns. *Eur J Personal* 34:733–752
- Scholbeck CA, Molnar C, Heumann C et al (2020) Sampling, intervention, prediction, aggregation: a generalized framework for model-agnostic interpretations. In: Cellier P, Driessens K (eds) *Machine learning and knowledge discovery in databases*. Springer, Cham, pp 205–216
- Schuwert T, Kaltefleiter LJ, Au JQ et al (2019) Enter the wild: autistic traits and their relationship to mentalizing and social interaction in everyday life. *J Autism Dev Disorders* 49:4193–4208
- Seedorff N, Brown G (2021) totalvis: a principal components approach to visualizing total effects in black box models. *SN Comput Sci* 2(3):1–12
- Servia-Rodríguez S, Rachuri KK, Mascolo C, et al (2017) Mobile sensing at the service of mental well-being: A large-scale longitudinal study. In: 26th international world wide web conference, WWW 2017. International World Wide Web Conferences Steering Committee, pp 103–112
- Shapley LS (1953) A value for n-person games. *Contrib Theory Games* 2(28):307–317
- Sharifzadeh S, Ghodsi A, Clemmensen LH et al (2017) Sparse supervised principal component analysis (sspsca) for dimension reduction and variable selection. *Eng Appl Artif Intell* 65:168–177

- Song L, Smola A, Gretton A et al (2012) Feature selection via dependence maximization. *J Mach Learn Res* 13:1393–1434
- Song L, Smola A, Gretton A, et al (2007) Supervised feature selection via dependence estimation. In: *Proceedings of the 24th international conference on Machine learning*, pp 823–830
- Stachl C, Hilbert S, Au JQ et al (2017) Personality traits predict smartphone usage. *Eur J Pers* 31(6):701–722
- Stachl C, Au Q, Schoedel R et al (2020a) Predicting personality from patterns of behavior collected with smartphones. *Proc Natl Acad Sci* 117:17680–17687
- Stachl C, Pargent F, Hilbert S et al (2020b) Personality research and assessment in the era of machine learning. *Eur J Personal* 34:613–631
- Strobl C, Boulesteix AL, Kneib T et al (2008) Conditional variable importance for random forests. *BMC Bioinform* 9:307
- Thomé S (2018) Mobile phone use and mental health; A review of the research that takes a psychological perspective on exposure. *Int J Environ Res Public Health* 15(12):2692
- Tibshirani R (1996) Regression shrinkage and selection via the lasso. *J Roy Stat Soc Ser B (Methodol)* 58(1):267–288
- Toloşi L, Lengauer T (2011) Classification with correlated features: unreliability of feature ranking and solutions. *Bioinformatics* 27(14):1986–1994
- Tripathi S, Hemachandra N, Trivedi P (2020) Interpretable feature subset selection: a Shapley value based approach. In: *Proceedings of 2020 IEEE international conference on big data, special session on explainable artificial intelligence in safety critical systems*
- Valentin S, Harkotte M, Popov T (2020) Interpreting neural decoding models using grouped model reliance. *PLOS Comput Biol* 16(1):e1007148
- Venables B, Ripley B (2002) *Modern applied statistics with S*
- Watson DS, Wright MN (2019) Testing conditional independence in supervised learning algorithms. [arXiv:1901.09917](https://arxiv.org/abs/1901.09917)
- Williamson BD, Gilbert PB, Simon NR, et al (2020) A unified approach for inference on algorithm-agnostic variable importance. [arXiv:2004.03683](https://arxiv.org/abs/2004.03683)
- Williamson B, Feng J (2020) Efficient nonparametric statistical inference on population feature importance using Shapley values. In: *International conference on machine learning*, PMLR, pp 10282–10291
- Witten D, Tibshirani R (2020) PMA: penalized multivariate analysis. *R Package Vers* 1(2):1
- Witten DM, Tibshirani R, Hastie T (2009) A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics* 10(3):515–534
- Wold S, Albano C, Dunn WJ et al (1984) *Multivariate data analysis in chemistry*. Springer, Dordrecht, pp 17–95
- Yarkoni T, Westfall J (2017) Choosing prediction over explanation in psychology: lessons from machine learning. *Perspect Psychol Sci* 12(6):1100–1122
- Yuan M, Lin Y (2006) Model selection and estimation in regression with grouped variables. *J R Stat Soc Ser B (Stat Methodol)* 68(1):49–67

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Part IV.

**Partitioning Approaches in Interpretable
Machine Learning**

5. REPID: Regional Effect Plots with implicit Interaction Detection

This is the first of three contributing articles addressing the second limitation stated in Section 1.1: *misleading interpretations of global explanations due to aggregation*. This work analyzes the aggregation bias caused by feature interactions for PD plots and one feature of interest. We suggest a new method called *regional effect plots with implicit interaction detection* (REPID), which is based on recursive partitioning to find interpretable regions in which feature interactions of the feature of interest are minimized. Thus, regional PD plots are more representative of the underlying observations within each region.

Contributing article: Herbinger, J., Bischl, B., and Casalicchio, G. (2022). Repid: Regional effect plots with implicit interaction detection. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, pp. 10209–10233, PMLR.

Author contributions: Julia Herbinger contributed to this paper as the first author with the following contributions:

Julia Herbinger and Giuseppe Casalicchio developed the project idea. Julia Herbinger developed the REPID method with continuous support and valuable input from Giuseppe Casalicchio. Julia Herbinger provided the mathematical foundation and proofs in the paper, which Giuseppe Casalicchio revised. Julia Herbinger designed and conducted the experiments. Julia Herbinger drafted the entire manuscript. All authors contributed to revisions of the paper. Giuseppe Casalicchio and Bernd Bischl gave valuable input throughout the project and suggested several notable modifications.

REPID: Regional Effect Plots with implicit Interaction Detection

Julia Herbinger
LMU Munich

Bernd Bischl
LMU Munich

Giuseppe Casalicchio
LMU Munich

Abstract

Machine learning models can automatically learn complex relationships, such as non-linear and interaction effects. Interpretable machine learning methods such as partial dependence plots visualize marginal feature effects but may lead to misleading interpretations when feature interactions are present. Hence, employing additional methods that can detect and measure the strength of interactions is paramount to better understand the inner workings of machine learning models. We demonstrate several drawbacks of existing global interaction detection approaches, characterize them theoretically, and evaluate them empirically. Furthermore, we introduce *regional effect plots with implicit interaction detection*, a novel framework to detect interactions between a feature of interest and other features. The framework also quantifies the strength of interactions and provides interpretable and distinct regions in which feature effects can be interpreted more reliably, as they are less confounded by interactions. We prove the theoretical eligibility of our method and show its applicability on various simulation and real-world examples.

1 INTRODUCTION

Many machine learning (ML) models are considered black-boxes, as they do not provide insights into how the model’s prediction function is composed and which features or interactions¹ are actually used by

¹Interactions describe to what extent a feature’s effect on the model prediction is influenced by other features.

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).

the model. This lack of transparency has been partially addressed by recent developments in the field of interpretable ML. In general, the literature distinguishes between local and global interpretation methods (Molnar et al., 2020). Global interpretation methods aim at explaining the overall behavior of an ML model. Examples include the partial dependence (PD) plot (Friedman, 2001), which visualizes the effect of a feature on the model’s prediction, and the permutation feature importance, which quantifies the relevance of features (Fisher et al., 2019). However, many of these global interpretation methods are confounded by feature interactions, meaning that they can produce misleading explanations when feature interactions are present because they often aggregate over individual effects of local interpretation methods and thereby obfuscate heterogeneous effects induced by feature interactions (Molnar et al., 2021b). This so-called *aggregation bias* (Mehrabi et al., 2021) is responsible for producing global explanations that are usually not representative or not valid for many individuals. Instead of explaining the ML model on a global level, local interpretation methods – such as individual conditional expectation (ICE) curves (Goldstein et al., 2015), LIME (Ribeiro et al., 2016), or Shapley values (Strumbelj and Kononenko, 2014) – can be used to understand how a feature influences an individual prediction. However, many local interpretation methods do not provide a global understanding of the ML model due to their local view (i.e., their explanations only refer to individual observations). Thus, it is often recommended to consider both local and global interpretation methods. For example, in the case of PD plots, looking additionally at ICE curves (Goldstein et al., 2015) can help to reveal interactions when the ICE curves are heterogeneous (see Figure 1). Yet, ICE curves are not able to quantify the strength of the underlying feature interactions, nor can they tell exactly which features interact with each other. On the other hand, other methods that quantify the interaction strength between features are available. However, they do not provide any visual component of how these interactions influence the effect of a feature of interest (Friedman et al., 2008; Greenwell et al., 2018). The

work in this paper is motivated by subgroup analysis (Su et al., 2009) as a trade-off between local and global explanations. We aim to uncover a possible *aggregation bias* in the PD plot by finding interpretable subgroups in the data with differing influences of a feature on the predictions. Hence, for well-performing ML models, this might also reveal a possible bias in the data (e.g., when the influence of a feature on the prediction strongly differs for certain subgroups, although it should not) and thus might be helpful to uncover possible negative societal impacts.

Contributions: We introduce *regional effect plots with implicit interaction detection* (REPID), a model-agnostic interpretation method that produces regional effect plots (REPs) in which feature effects are less confounded by interactions. Regions are obtained by a decision tree and thus represent interpretable and distinct subgroups in the feature space. We also propose a new measure to detect and quantify interactions with a feature of interest, which can be used to rank interactions according to their strength. To receive a broader and more competitive comparability, we derive another global interaction index based on SHAP interaction values (Lundberg et al., 2018). We mathematically prove the theoretical meaningfulness of our method and demonstrate its advantages compared to not only the well-known H-statistic (Friedman et al., 2008), but also to Greenwell’s interaction index (Greenwell et al., 2018) and our derived global SHAP interaction index. Finally, we demonstrate the usefulness of our method on real-world data.

Open Science: The implementation of the proposed method and the fully reproducible code for all experiments are provided in a public repository².

2 BACKGROUND AND RELATED WORK

Notation: Consider a p -dimensional feature space $\mathcal{X} \in \mathbb{R}^p$ and a target space \mathcal{Y} (e.g., $\mathcal{Y} = \mathbb{R}$ for regression). The corresponding random variables are $X = (X_1, \dots, X_p)$ for the features and Y for the target. ML algorithms learn a prediction model \hat{f} using training data $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$ sampled i.i.d. from the unknown joint distribution $\mathbb{P}_{X,Y}$. In our notation, the i -th observation is denoted by $\mathbf{x}^{(i)} = (x_1^{(i)}, \dots, x_p^{(i)})^T$, and $\mathbf{x}_j = (x_j^{(1)}, \dots, x_j^{(n)})^T$ denotes the realizations of the j -th feature X_j .

PD Plot (Friedman, 2001): The marginal relationship of features on model predictions can be visual-

ized by PD plots. Consider a set of feature indices $S \subseteq \{1, \dots, p\}$ and its complement $C = S^c$. Each observation $\mathbf{x}^{(i)}$ can be partitioned into $\mathbf{x}_S^{(i)}$ and $\mathbf{x}_C^{(i)}$ containing only features indexed by S and C , respectively. X_S and X_C refer to the corresponding random variables. The PD function of features indexed by S marginalizes over features in C and is defined as $f_S^{PD}(\mathbf{x}_S) = E_{X_C}[\hat{f}(\mathbf{x}_S, X_C)]$. The PD function is estimated by Monte-Carlo integration:

$$\hat{f}_S^{PD}(\mathbf{x}_S) = \frac{1}{n} \sum_{i=1}^n \hat{f}(\mathbf{x}_S, \mathbf{x}_C^{(i)}). \quad (1)$$

Here, $\hat{f}(\mathbf{x}_S, \mathbf{x}_C^{(i)})$ can be read as the prediction of the i -th observation where features in S were replaced by \mathbf{x}_S . Plotting the pairs $\{(\mathbf{x}_S^{(k)}, \hat{f}_S(\mathbf{x}_S^{(k)}))\}_{k=1}^m$ using grid points³ denoted by $\mathbf{x}_S^{(1)}, \dots, \mathbf{x}_S^{(m)}$ yields a PD curve. The mean-centered PD function can be estimated by

$$\hat{f}_S^{PD,c}(\mathbf{x}_S) = \hat{f}_S^{PD}(\mathbf{x}_S) - \frac{1}{m} \sum_{k=1}^m \hat{f}_S^{PD}(\mathbf{x}_S^{(k)}).$$

If $|S| = 2$, we get a 2-dimensional PD plot showing the joint marginal effect of the 2 features included in S .

ICE Plot (Goldstein et al., 2015): The averaging in Eq. (1) can obfuscate complex relationships resulting from feature interactions. ICE plots address this problem by directly visualizing individual curves for each observation, i.e., $\{(\mathbf{x}_S^{(k)}, \hat{f}_S(\mathbf{x}_S^{(k)}, \mathbf{x}_C^{(i)}))\}_{k=1}^m$ for all $i \in \{1, \dots, n\}$. ICE curves will usually have different shapes if interactions with other features in C are present. To facilitate the visual identification of heterogeneous ICE curves and, consequently, the presence of interactions, the authors propose the derivative-ICE (d-ICE) plot. Assuming that there are no interactions between features \mathbf{x}_S and \mathbf{x}_C , the prediction function can be written as $\hat{f}(\mathbf{x}) = \hat{f}(\mathbf{x}_S, \mathbf{x}_C) = g(\mathbf{x}_S) + h(\mathbf{x}_C)$. Hence, the partial derivatives of all ICE curves $\frac{\delta \hat{f}(\mathbf{x}_S, \mathbf{x}_C^{(i)})}{\delta \mathbf{x}_S} = g'(\mathbf{x}_S)$ do not depend on $\mathbf{x}_C^{(i)}$, which means that d-ICE curves have the same shape if there are no interactions. The d-ICE plot visualizes the partial derivatives of ICE curves along with their standard deviation to highlight regions in \mathbf{x}_S where the d-ICE curves are heterogeneous (see Figure 1).

Visual INteraction Effects (VINE) (Britton, 2019): The principle of VINE is to cluster similar slopes of ICE curves to obtain clusters where the curves are less affected by interactions based on a three-step approach: (1) for a feature of interest, find clusters where the ICE curves of that feature have similar slopes using, e.g., agglomerative clustering, (2) for each found cluster, create a binary label containing the information of whether an observation belongs to the considered cluster or any other cluster and apply a tree

²<https://github.com/JuliaHerbinger/repid>

³Common choices are randomly selected feature values, quantiles, or equidistant values (Molnar et al., 2021b).

Julia Herbinger, Bernd Bischl, Giuseppe Casalicchio

stump, (3) identify the split feature and its split point and merge clusters that use the same feature and a similar split point. Although VINE is based on a similar strategy as our approach, its three-step approach has several disadvantages (see Section 3.1.1). Approaches to group ICE curves to reduce feature dependencies instead of feature interactions is introduced in Molnar et al. (2021a) and Grömping (2020).

H-Statistic (Friedman et al., 2008): The H-Statistic is based on the assumption that if two features do not interact, the 2-dimensional mean-centered PD function of two features \mathbf{x}_j and \mathbf{x}_l is additively separable and can be decomposed into the sum of their mean-centered 1-dimensional PDs, i.e.,

$$f_S^{PD,c}(\mathbf{x}_S) = f_j^{PD,c}(\mathbf{x}_j) + f_l^{PD,c}(\mathbf{x}_l), \text{ with } S = \{j, l\}.$$

The stronger an interaction effect, the more the sum of $f_j^{PD,c}(\mathbf{x}_j)$ and $f_l^{PD,c}(\mathbf{x}_l)$ will deviate from $f_S^{PD,c}(\mathbf{x}_S)$. Hence, the H-statistic computes the interaction strength between two features \mathbf{x}_j and \mathbf{x}_l by quantifying the degree of this deviation using

$$\hat{H}_S^2 = \frac{\sum_{i=1}^n (f_S^{PD,c}(\mathbf{x}_S^{(i)}) - \sum_{k \in S} f_k^{PD,c}(\mathbf{x}_k^{(i)}))^2}{\sum_{i=1}^n (f_S^{PD,c}(\mathbf{x}_S^{(i)}))^2}. \quad (2)$$

Greenwell's interaction index (Greenwell et al., 2018): The interaction strength between two features \mathbf{x}_j and \mathbf{x}_l is quantified based on the variability of the PD function of \mathbf{x}_j conditioned on a fixed value of \mathbf{x}_l (see Appendix A.3.1).

However, the H-Statistic and the Greenwell's interaction index only quantify interaction effects and do not visualize how interactions influence the marginal effect of a feature. Moreover, both methods are sensitive to varying main effects (see Section 3.2.1 and 4.1).

Functional ANOVA (fANOVA) (Hooker, 2004): The fANOVA decomposes the prediction function as follows:

$$\hat{f}(\mathbf{x}) = g_0 + \sum_{k=1}^p \sum_{W \subseteq \{1, \dots, p\}, |W|=k} g_W(\mathbf{x}_W) \quad (3)$$

where $E_X[g_W(\mathbf{x}_W)] = 0$ for all feature index sets W (zero-means property). While $g_W(X_W)$ with $|W| = 1$ refers to main (or *first-order*) effects, $g_W(X_W)$ with $|W| > 1$ refers to interactions (or *higher-order*) effects. Based on the decomposition in Eq. (3), the authors detect interactions of any order by applying an efficient search algorithm and visualize them in an interaction network graph. However, the network only shows the presence of feature interactions and does not quantify the interaction strength or illustrate how they influence the prediction. A discussion on the assumptions and application of the fANOVA decomposition in the context of this paper is provided in Appendix A.1.

SHAP interaction values (Lundberg et al., 2018): The method is based on Shapley values (Shapley, 1953) and Shapley interaction indices (Fujimoto et al., 2006) from game theory. In the ML context, SHAP interaction values of two features quantify the pure interaction effect after accounting for the individual feature effects. The SHAP interaction values separate the interaction effect from the main effects of two features indexed by j and l (for $j \neq l$) for an observation \mathbf{x} :

$$\Phi_{j,l}(\mathbf{x}) = \sum_{S \subseteq \{1, \dots, p\} \setminus \{j,l\}} \frac{|S|!(p-|S|-2)!}{2^{(p-1)!}} \nabla_{j,l}(\mathbf{x}_S),$$

where $\nabla_{j,l}(\mathbf{x}_S) = f_{S \cup \{j,l\}}^{PD}(\mathbf{x}_{S \cup \{j,l\}}) - f_{S \cup \{j\}}^{PD}(\mathbf{x}_{S \cup \{j\}}) - f_{S \cup \{l\}}^{PD}(\mathbf{x}_{S \cup \{l\}}) + f_S^{PD}(\mathbf{x}_S)$. The SHAP interaction values have only been introduced on an observational level, where the final plot over all observations shows the influence of the interaction effect on the prediction.

3 THE REPID METHOD

REPID visualizes regional marginal effects of a certain feature of interest \mathbf{x}_S with $|S| = 1$ depending on its interactions with other features and quantifies the underlying interaction strength. The following simulation example demonstrates the benefits of our method compared to existing ones. We draw $n = 500$ samples for 6 independent random variables, which are distributed as follows: $X_1, X_2 \sim \mathcal{U}(-1, 1)$, $X_3, X_5 \sim \mathcal{B}(n, 0.5)$, $X_4 \sim \mathcal{B}(n, 0.7)$ and $X_6 \sim \mathcal{N}(1, 5)$. The true relationship is described by

$$f(\mathbf{x}) = 0.2\mathbf{x}_1 - 8\mathbf{x}_2 + 8\mathbf{x}_2 \mathbb{1}_{(\mathbf{x}_1 > 0)} + 16\mathbf{x}_2 \mathbb{1}_{(\mathbf{x}_3 = 0)} + \epsilon \quad (4)$$

with $\epsilon \sim \mathcal{N}(0, 1)$. We fit a random forest (RF) with 500 trees on the data. Due to the linear relationship, we can assume that the interaction strength between \mathbf{x}_2 and \mathbf{x}_3 is higher than the one between \mathbf{x}_2 and \mathbf{x}_1 .

3.1 Regional Effect Plots

3.1.1 Motivation

PD plots are often shown together with their underlying ICE curves (see Figure 1). The heterogeneous shapes of ICE curves imply the presence of feature interactions. Although ICE or d-ICE plots indicate interactions, they do not provide any information on which other features are responsible for these interactions and how the underlying interaction influences the marginal effect of \mathbf{x}_S (see Figure 1). Grouping homogeneous ICE curves will reduce the presence of individual interaction effects within a group. This leads to regional PD plots that actually reflect the pure marginal effect of \mathbf{x}_S within this group. VINE (Britton, 2019) implements this idea by clustering ICE curves with similar slopes (see Section 2). However, VINE is only

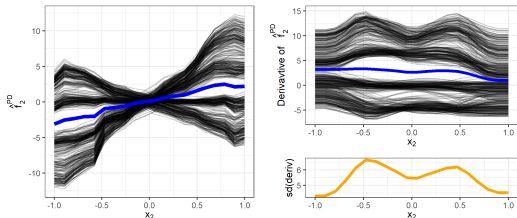


Figure 1: Left: ICE curves (black) and PD plot (blue) for x_2 . Right: Smoothed d-ICE curves (upper plot) and standard deviation of d-ICE curves (lower plot).

a visual tool and does not quantify or rank feature interactions. Furthermore, VINE is an unsupervised approach, and its solution depends on the number of clusters k that must be chosen (which is not trivial). Another drawback is that VINE “finds” feature interactions in an inconvenient second step by fitting a separate tree stump for each cluster (see Section 2). Due to the different tree stumps used in VINE, the derived decision rules are often not distinct and therefore difficult to interpret. In a third step, VINE introduces a post-hoc merging of clusters based on similar decision rules. In Figure 2, we show that this three-step approach does not always lead to meaningful groupings. While in the left plot, the ICE curves are divided meaningfully into 2 clusters based on the most interacting feature x_3 (according to Eq. (4)), the clusters in the right plot do not divide the ICE curves into visually meaningful groups with homogeneous ICE curves.

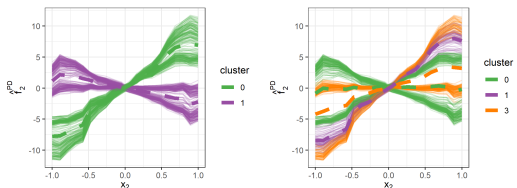


Figure 2: ICE and regional PD (dashed) plot of x_2 clustered by VINE for $k = 2$ (left) and $k = 5$ (right). The 5 clusters are reduced to 3 by post-hoc merging. Cluster numbers 0 and 3 still contain differing individual interaction effects, which are averaged and hence not represented well by the regional PD plot.

3.1.2 Methodology

Here, we derive a new tree-based approach to determine optimal REPs for any feature of interest x_S . REPs are regional PD plots that aggregate ICE curves

within automatically identified regions where feature effects are less confounded by interactions. Our aim is to recursively split the entire data referred to by index set $\mathcal{N} = \{1, \dots, n\}$ into interpretable regions to obtain more homogeneous ICE curves for x_S within the split regions denoted by \mathcal{N}_g (where $g \in \{1, \dots, G\}$ indexes a certain node of the tree and G is the number of all tree nodes). Hence, we want to split \mathcal{N} in such a way that ICE curves within the obtained regions have a similar shape, meaning that the distance of these ICE curves to the REP estimate (i.e., $\hat{f}_{S|\mathcal{N}_g}^{PD}(x_S) := \frac{1}{|\mathcal{N}_g|} \sum_{i \in \mathcal{N}_g} \hat{f}(x_S, x_C^{(i)})$) is small. To that end, we propose a tree-based partitioning in Algorithm 1, which refers only to a single binary split and is inspired by the CART algorithm (Breiman et al., 1984)⁴. The splitting is recursively repeated until the split criterion (denoted by $\mathcal{I}(\hat{t}, \hat{j})$ in Algorithm 1) does not improve anymore compared to the previous split or until a pre-specified stop criterion is met. The split criterion is based on a suitable risk function \mathcal{R} that operates on ICE curves (see also Eq. (6)).

Algorithm 1: Tree-based Partitioning

input: index set \mathcal{N} , risk \mathcal{R}_{L2} (see, e.g., Eq. (6))

output: child nodes $\mathcal{N}_l^{t,j}$ and $\mathcal{N}_r^{t,j}$

for each feature indexed by $j \in C$ **do**

for every split t on feature x_j **do**

$$\mathcal{N}_l^{t,j} = \{i \in \mathcal{N}\}_{x_j^{(i)} \leq t}; \mathcal{N}_r^{t,j} = \{i \in \mathcal{N}\}_{x_j^{(i)} > t}$$

$$\mathcal{I}(t, j) = \mathcal{R}_{L2}(\mathcal{N}_l^{t,j}) + \mathcal{R}_{L2}(\mathcal{N}_r^{t,j})$$

end for

end for

Choose $\hat{t}, \hat{j} \in \arg \min_{t,j} \mathcal{I}(t, j)$

We first estimate the mean-centered ICE curves by $\hat{f}^c(x_S, x_C^{(i)}) = \hat{f}(x_S, x_C^{(i)}) - \frac{1}{m} \sum_{k=1}^m \hat{f}(x_S^{(k)}, x_C^{(i)})$. Since we want to minimize the shape differences between ICE curves in the regions, we then define the risk function \mathcal{R}_{L2} in Eq. (6)⁵ such that the variance (L2 loss) of the mean-centered ICE curves is minimized. This can be estimated by calculating the L2 loss of the mean-centered ICE curves at each grid point (see Eq. (5)) and aggregating it over all grid points:

$$\mathcal{L}(\mathcal{N}_g, x_S) = \sum_{i \in \mathcal{N}_g} \left(\hat{f}^c(x_S, x_C^{(i)}) - \hat{f}_{S|\mathcal{N}_g}^{PD,c}(x_S) \right)^2 \quad (5)$$

$$\mathcal{R}_{L2}(\mathcal{N}_g) = \sum_{k=1}^m \mathcal{L}(\mathcal{N}_g, x_S^{(k)}) \quad (6)$$

⁴Algorithm 1 is defined for numerical features. For categorical features, we use an exhaustive search as seen in CART. The computational feasibility of this procedure depends on the number of categories.

⁵Multiplying with $\frac{1}{m}$ to obtain the average loss can be neglected for optimization.

Julia Herbinger, Bernd Bischl, Giuseppe Casalicchio

Theorem 1 If Eq. (3) holds, then $\hat{f}^c(\mathbf{x}_S, \mathbf{x}_C^{(i)})$ with $|S| = 1$ can be decomposed into the mean-centered⁶ main effect of \mathbf{x}_S (i.e. $g_S^{cS}(\mathbf{x}_S)$) and the mean-centered interaction effect of \mathbf{x}_S with \mathbf{x}_C for the i -th observation (i.e., $g_{C_k \cup \{S\}}^{cS}(\mathbf{x}_S, \mathbf{x}_{C_k}^{(i)})$):

$$\hat{f}^c(\mathbf{x}_S, \mathbf{x}_C^{(i)}) = g_S^{cS}(\mathbf{x}_S) + \sum_{k=1}^{p-1} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k \cup \{S\}}^{cS}(\mathbf{x}_S, \mathbf{x}_{C_k}^{(i)}).$$

Corollary 1.1 If Eq. (3) holds, then $f_S^{PD,c}(\mathbf{x}_S) = E_{X_C}[\hat{f}^c(\mathbf{x}_S, X_C)]$ with $|S| = 1$ can be decomposed into

$$g_S^{cS}(\mathbf{x}_S) + \sum_{k=1}^{p-1} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} E_{X_C} \left[g_{C_k \cup \{S\}}^{cS}(\mathbf{x}_S, X_{C_k}) \right].$$

The proof can be found in Appendix A.1.1.

Based on Theorem 1 and Corollary 1.1 – where we show that the mean-centered ICE curves and PD function can be decomposed in first-order and higher-order terms which depend on \mathbf{x}_S – we can prove in Theorem 2, that our risk function of Eq. (6) only depends on the interaction effects between \mathbf{x}_S and features in \mathbf{x}_C . Hence, by minimizing this risk function, we minimize the individual interaction effects between the feature of interest and all other features. Thus, we minimize the shape differences between ICE curves in each region. Theorem 3 states that the theoretical minimum of our split criterion leads to the optimal solution we aim to achieve, meaning that for each final region, all ICE curves are best represented by the REP.

Theorem 2 The distance minimized by the risk function \mathcal{R}_{L_2} of Eq. (6) only depends on the mean-centered interaction effects between \mathbf{x}_S with $|S| = 1$ and all features interacting with \mathbf{x}_S , i.e., for the i -th observation, the distance results in

$$\sum_{k=1}^{p-1} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k \cup \{S\}}^{cS}(\mathbf{x}_S, \mathbf{x}_{C_k}^{(i)}) - E_{X_C} [g_{C_k \cup \{S\}}^{cS}(\mathbf{x}_S, X_{C_k})].$$

The proof can be found in Appendix A.1.2.

Theorem 3 If $\mathcal{I}(t, j) = 0$, i.e., the theoretical minimum of the split criterion is reached for a split, then the ICE curves within each of the child nodes \mathcal{N}_l and \mathcal{N}_r are identical to the respective REP (e.g., $\hat{f}^c(\mathbf{x}_S, \mathbf{x}_C^{(i)}) = \hat{f}_{S|\mathcal{N}_i}^{PD,c}(\mathbf{x}_S) \quad \forall i \in \mathcal{N}_i$).

Proof 3 Since $\mathcal{R}_{L_2}(\mathcal{N}_g) \geq 0 \quad \forall g \in \{1, \dots, G\}$, $\mathcal{I}(t, j) = 0$ implies $\hat{f}^c(\mathbf{x}_S, \mathbf{x}_C^{(i)}) = \hat{f}_{S|\mathcal{N}_g}^{PD,c}(\mathbf{x}_S)$,

⁶ $g_W^{cS}(X_W) = g_W(X_W) - E_{X_S}[g_W(X_W)]$ is the mean-centered counterpart of $g_W(X_W)$ of Eq. (3) regarding X_S .

$\forall i \in \mathcal{N}_g, \forall g \in \{l, r\}$.

Applying our method to the simulation example introduced at the beginning of Section 3.1 leads to the REPs shown in Figure 3 after two splits. The first binary split divides the ICE curves of \mathbf{x}_2 using feature \mathbf{x}_3 , which interacts most with \mathbf{x}_2 (according to Eq.(4)). Each of the 2 resulting regions is then split again into 2 groups by feature \mathbf{x}_1 , which also interacts with \mathbf{x}_2 . Hence, after the second split, we receive interpretable and distinct regions with REPs that represent each sub-population well.

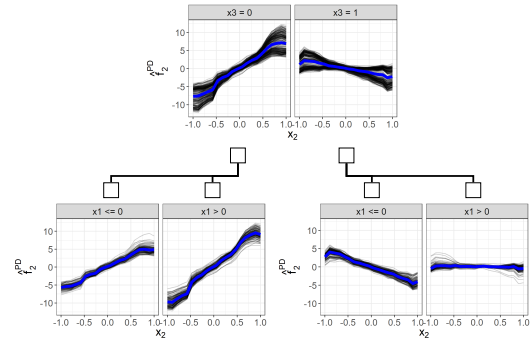


Figure 3: ICE curves for \mathbf{x}_2 grouped by REPID (black) and REPs (blue).

3.2 Quantifying Interaction Strength

3.2.1 Motivation

Besides understanding how other features influence the marginal effect of \mathbf{x}_S , users might be interested in how strong these interactions are and how to rank these features regarding their interaction strength with \mathbf{x}_S . The H-Statistic defined in Section 2 is a global measure that quantifies the strength of interaction between two features. However, its values are influenced by the main effects of the two regarded features (see Theorem 4). Hence, the two-way interaction with the highest H-Statistic value is not necessarily the strongest interaction, which we demonstrate in Section 4.1.

Theorem 4 The variance of the 2-dimensional mean-centered PD plot of features \mathbf{x}_j and \mathbf{x}_l ($\text{Var}(f_S^{PD,c}(\mathbf{x}_S))$ with $S = \{j, l\}$) depends on the mean-centered main effects (i.e., $g_j^{cS}(\mathbf{x}_j)$ and $g_l^{cS}(\mathbf{x}_l)$) of the two features of interest \mathbf{x}_j and \mathbf{x}_l . Since $\text{Var}(f_S^{PD,c}(\mathbf{x}_S))$ is the denominator of the H-Statistic, which is estimated as in Eq. (2), the H-Statistic itself also depends on the main effects of features in S . The proof can be found in Appendix A.1.3.

The global interaction index proposed by Greenwell et al. (2018) suffers from the same problem that we illustrate in Section 4.1 (see also Appendix A.3.1). A third method of quantifying the two-way interaction strength between features is based on SHAP interaction values (see Section 2). To the best of our knowledge, SHAP interaction values have been only defined on an observational level. Similar to the global feature importance used in Lundberg et al. (2018) to rank features according to their global impact in their SHAP summary plots, we suggest summarizing the individual SHAP interaction values for two features \mathbf{x}_j and \mathbf{x}_l into a global SHAP interaction index by

$$I_{j,l}^{\text{rel}} = \frac{I_{j,l}}{\sum_{l \in \{1, \dots, p\} \setminus \{j\}} I_{j,l}} \text{ where } I_{j,l} = \sum_{i=1}^n |\Phi_{j,l}(\mathbf{x}^{(i)})|.$$

Since the absolute values $I_{j,l}$ are difficult to interpret, we prefer a relative version $I_{j,l}^{\text{rel}}$, which we call the SHAP interaction index and can be interpreted as the proportion of all two-way interactions with \mathbf{x}_j to which the l -th feature contributes. By definition, SHAP interaction values only contain the interaction effect between \mathbf{x}_j and \mathbf{x}_l . Hence, in contrast to the H-Statistic, varying main effects do not change the ranking of our proposed global SHAP interaction index $I_{j,l}^{\text{rel}}$. However, both SHAP interaction indices and the H-Statistic are based on the joint distribution of the two regarded features, and hence, correlations between \mathbf{x}_S and features in \mathbf{x}_C might bias the interaction value calculated by these methods, as demonstrated in Section 4.1.

3.2.2 Methodology

Here, We derive an interaction index based on the split criterion minimized in Algorithm 1 and Eq. (6), and we prove its advantages compared to alternatives mentioned in Section 3.2.1. Since the risk function of our split criterion is based on the variance of mean-centered ICE curves – which measures the degree of existing feature interactions with \mathbf{x}_S – we can use the achieved risk reduction after a split to quantify the interaction strength. For better interpretability and comparability, we define the relative interaction importance for each parent node \mathcal{N}_P by

$$\text{intImp}(\mathcal{N}_P) = \frac{\mathcal{R}_{L_2}(\mathcal{N}_P) - (\mathcal{R}_{L_2}(\mathcal{N}_l) + \mathcal{R}_{L_2}(\mathcal{N}_r))}{\mathcal{R}_{L_2}(\mathcal{N})} \quad (7)$$

with $l, r \in \{1, \dots, G\}$ denoting the left and right child node of a parent node \mathcal{N}_P and \mathcal{N} representing the root node. Hence, $\text{intImp}(\mathcal{N}_P)$ measures the relative risk reduction after splitting \mathcal{N}_P compared to the risk within the root node $\mathcal{R}_{L_2}(\mathcal{N})$. Let $\mathcal{B}_P \subset \{1, \dots, G\}$ denote the index set of all parent nodes (i.e., all nodes that have child nodes), and let $\mathcal{B}_j \subseteq \mathcal{B}_P$ denote the subset of these parent nodes that used the regarded

feature \mathbf{x}_j for splitting. To obtain the relative interaction importance of feature \mathbf{x}_j , we sum up the relative interaction importance over the parent nodes in \mathcal{B}_j :

$$\text{intImp}_j = \sum_{P \in \mathcal{B}_j} \text{intImp}(\mathcal{N}_P). \quad (8)$$

This principle of summing up the relative risk reduction of individual splits regarding a certain feature in order to measure the interaction strength is related to how a decision tree measures the Gini or mean decrease impurity (MDI) feature importance (Breiman et al., 1984). We obtain a measure that reports how important each of these features is for reducing interactions and thus obtaining more representative REPs for \mathbf{x}_S . Our proposed interaction importance in Eq. (8) only depends on the interaction effects between \mathbf{x}_j and \mathbf{x}_S and not on their main effects (see Theorem 2), as opposed to the H-Statistic or the interaction index of Greenwell et al. (2018). Furthermore, we show by Theorem 5 that intImp – in contrast to the H-Statistic and the SHAP interaction index $I_{j,l}^{\text{rel}}$ – is not influenced by correlations between \mathbf{x}_S and \mathbf{x}_j .

Theorem 5 *Correlations between X_S and X_C do not influence the splitting procedure of REPID, since the loss function \mathcal{L} of Eq. (5) does not contain a covariance term between X_S and features in X_C . The proof can be found in Appendix A.1.4.*

To determine how well the resulting REPs in the terminal nodes represent the underlying ICE curves, we derive an R^2 measure, which is commonly used in statistics. The R^2 can be calculated by $R^2 = 1 - \frac{\text{SSE}(\text{complex model})}{\text{SSE}(\text{baseline model})}$ where the baseline model is, e.g., a constant mean prediction and the SSE is the sum of squared errors of the model. The measure (usually) only takes values between 0 and 1 when applied on training data. While a value of 1 indicates that the complex model fits the data perfectly, a value of 0 implies that the complex model does not outperform the baseline model. Similar to this concept, we use the global PD plot as our baseline model. Our complex model is the additive combination of the REPs in the terminal nodes of the final tree. Hence, each additive functional component (REP) is only valid for the specified region. The SSE of each model is measured by the variability of the underlying ICE curves. Let $\mathcal{B}_t = \mathcal{B}_P^c$ denote the subset of terminal nodes in a symmetric tree. We derive an interaction-related R^2 measure by aggregating the interaction importance over all parent nodes \mathcal{B}_P :

$$R_{\text{int}}^2 = \sum_{P \in \mathcal{B}_P} \text{intImp}(\mathcal{N}_P) = 1 - \frac{\sum_{t \in \mathcal{B}_t} \mathcal{R}_{L_2}(\mathcal{N}_t)}{\mathcal{R}_{L_2}(\mathcal{N})} \quad (9)$$

A detailed derivation can be found in Appendix A.2.

For our example, we obtain the relative interaction importance values for \mathbf{x}_2 , as stated in Table 1. Since

Julia Herbinger, Bernd Bischl, Giuseppe Casalicchio

both child nodes after the first split use \mathbf{x}_1 as the splitting feature, the relative interaction importance values of the two nodes can be aggregated to obtain $intImp_1 = 0.14$. It follows that REPID detects (only) the feature interactions with \mathbf{x}_2 that have been specified in the underlying data-generating process and also ranks them in the correct order. The total variance after the second split is reduced by $R_{int}^2 = 97.5\%$ compared to the root node, suggesting that resulting REPs are now meaningful representatives for the average marginal effect, as shown in Figure 3.

Table 1: Relative interaction importance on a node level (left) and on a feature level (right). Gray shadings indicate how $intImp_j$ is calculated from $intImp(\mathcal{N}_P)$. The parameters d and P indicate the tree depth and the index of the parent node, respectively.

d	P	\mathbf{x}_j	$intImp(\mathcal{N}_P)$	\mathbf{x}_j	$intImp_j$
0	1	\mathbf{x}_3	0.835	\mathbf{x}_3	0.835
1	2	\mathbf{x}_1	0.074	\mathbf{x}_1	0.14
1	3	\mathbf{x}_1	0.066		

Stop Criteria A possible stop criterion for the tree is to limit the maximum depth of the tree or to define a minimum number of observations for each node. Furthermore, we can apply a stop criterion based on the interaction importance $intImp$. Let \mathcal{N}_g be the node we want to split and let \mathcal{N}_P be its parent node. Then, we only split deeper if $intImp(\mathcal{N}_g) \geq \gamma \cdot intImp(\mathcal{N}_P)$, with $\gamma \in [0, 1]$. In other words, we only split deeper if the improvement of the current split is at least as large as a pre-specified proportion of the improvement of the previous split. The suggested criteria can also be combined and the hyperparameters must be chosen by the user and usually depend on the underlying setting.

4 SIMULATION EXAMPLES

For many model-agnostic interpretation techniques – including interaction detection methods – ground truth information is usually not available on real-world data. Therefore, well-constructed simulation experiments with a known ground truth are often used for empirical evaluations and comparisons, while only one or few real-world datasets are used to demonstrate practical applicability (e.g., see Friedman et al. (2008), Fisher et al. (2019), Goldstein et al. (2015), Greenwell et al. (2018), or Aas et al. (2021)). Hence, we follow this commonly used approach to evaluate our method using various simulation settings.

4.1 Weaknesses of other Methods

In Section 3.2.1, we described disadvantages of several interaction measures from a theoretical perspective. In the following simulation example, we provide further empirical evidence. To be able to modify the degree of the feature dependencies later on, we use a Gaussian copula to simulate the data in all settings. In the initial setting, we draw 1000 samples of four approximately i.i.d. random variables, which are marginally $X_1, \dots, X_4 \sim \mathcal{U}(-1, 1)$, and assume the true underlying function of $f(\mathbf{x}) = r(\mathbf{x}) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, (\sigma(r(\mathbf{x})) \cdot 0.1)^2)$. We define the remainder by $r(\mathbf{x}) = \sum_{j=1}^4 \mathbf{x}_j + \mathbf{x}_1 \mathbf{x}_2 + \mathbf{x}_2 \mathbf{x}_3 + \mathbf{x}_1 \mathbf{x}_3 + \mathbf{x}_1 \mathbf{x}_2 \mathbf{x}_3$. To avoid undefined interaction effects, we fit a correctly specified linear model on the data. We repeat the experiment 30 times, and each time, we measure the interaction strength between \mathbf{x}_2 and the other three features using REPID as well as the three alternatives (the H-statistic, the Greenwell’s interaction index, and the SHAP interaction index). On three adjusted settings, we then illustrate that already small modifications of main effect sizes or feature dependencies may produce misleading results for some of the alternatives when used as a measure to rank interactions, while REPID provides correct and stable results. For the computations, we used an equidistant grid of size 20 for REPID and Greenwell’s interaction index. For better comparability, we used a sample size of 20 for the H-Statistic. We calculated the SHAP interaction index by aggregating the individual interaction indices for 100 randomly sampled observations, which are approximated by using 20 random permutations for all possible feature coalitions. For REPID, we combine the stop criteria described in Section 3.2.2 as follows: We use a maximum depth of 6, a minimum number of 10 observations per node, and an improvement factor of $\gamma = 0.15$.

(1) *Initial Setting:* The plot on the top left of Figure 4 shows that, for the initial setting, all methods on average correctly assign the same interaction importance to \mathbf{x}_1 as to \mathbf{x}_3 , while \mathbf{x}_4 does not interact with \mathbf{x}_2 .

(2) *Small main effects:* If we reduce the main effect of \mathbf{x}_1 to 0.1, we observe in the top right plot of Figure 4 that its interaction strength with \mathbf{x}_2 increases on average when the H-Statistic is used. This effect can be explained by Theorem 4. Hence, when main effects decrease, the proportion of the variance that explains the interaction between \mathbf{x}_1 and \mathbf{x}_2 increases compared to the proportion of the variance that explains the respective main effects. Also the method of Greenwell’s interaction index depends on the main effect sizes. However, since Greenwell’s interaction index includes the main effects in the nominator, the effect on the resulting interaction index is opposite to the one of the H-Statistic which includes the main effects

in the denominator. On the other hand, the SHAP interaction index as well as REPID are only based on interaction effects, and hence, varying main effects do not change the ranking. The plot on the bottom right of Figure 4 illustrates how problematic small main effects can be when the H-Statistic is applied. The H-Statistic leads to average interaction values close to 1 for \mathbf{x}_1 and \mathbf{x}_3 , although the actual interaction effect of \mathbf{x}_1 with \mathbf{x}_2 is twice as high as that of \mathbf{x}_3 with \mathbf{x}_2 .

(3) *Dependencies between the feature of interest and other features:* In the lower left plot of Figure 4, the correlation between \mathbf{x}_1 and \mathbf{x}_2 has been set to $\rho_{12} \approx 0.9$. Since we face a positive linear interaction effect between \mathbf{x}_1 and \mathbf{x}_2 , a positive linear correlation between these features leads to an increasing denominator of the H-Statistic. Hence, the respective H-Statistic value decreases compared to features that are independent of \mathbf{x}_2 (here, \mathbf{x}_3). The SHAP interaction index for \mathbf{x}_1 is higher than for \mathbf{x}_3 , since in this case, it can be shown that the interaction strength is an additive combination of the interaction effect and the covariance of the interacting features. Conversely, Greenwell’s interaction index is based on the variance of conditional marginal effects, and hence, the interaction index is not influenced by dependencies between the feature of interest and other features. The same holds for REPID, as proven with Theorem 5.

A summary of the simulation settings and key results is provided in Appendix B.1. Detailed theoretical derivations and explanations can be found in Appendix A.3.

4.2 Comparison on More Complex Settings

The aim in this simulation is to show that REPID detects existing interactions correctly in a more complex non-linear setting and to compare the results to the H-Statistic. Analogous to Hu et al. (2020), we draw 2000 samples of 10 independently and uniformly distributed random variables $X_1, \dots, X_{10} \sim \mathcal{U}(-1, 1)$ and assume the following true underlying function:

$$\begin{aligned} f(\mathbf{x}) = & 6\mathbf{x}_1 + \mathbf{x}_2^2 - \pi\mathbf{x}_3 + \exp^{-2\mathbf{x}_4} + (2 + |\mathbf{x}_5|)^{-1} \\ & + \mathbf{x}_6 \log |\mathbf{x}_6| + 2\mathbf{x}_3 \mathbb{1}_{(\mathbf{x}_1 > 0)} \mathbb{1}_{(\mathbf{x}_2 > 0)} + 2\mathbf{x}_2 \mathbb{1}_{(\mathbf{x}_4 > 0)} \\ & + 4(\mathbf{x}_2 \mathbb{1}_{(\mathbf{x}_2 > 0)})^{|\mathbf{x}_6|} + |\mathbf{x}_2 + \mathbf{x}_8| + \epsilon \end{aligned}$$

with $\epsilon \sim \mathcal{N}(0, 0.25)$. Hence, \mathbf{x}_2 interacts with five other features in a more complex and non-linear way. To avoid undefined interaction effects in a fitted model, we fit a correctly specified generalized additive model (GAM) and a tree-based extreme gradient boosting model (XGBOOST) with correctly specified interaction constraints⁷, a learning rate of 0.1, a maximum

⁷The “xgboost” library (Chen and Guestrin, 2016) enables definition of which features are allowed to interact with each other.

number of iterations of 1000, and a maximum tree depth of 6 on the simulated data. The performance of each model is measured by a separately simulated test set with the same distributional assumptions of size 100,000 and is reported in Figure 5. We repeat the experiment 30 times, and each time, we measure the interaction strength between \mathbf{x}_2 and the other nine features using REPID and the H-Statistic. For both methods, we again use a grid size of 20. For REPID, we apply the same stop criteria as in Section 4.1 but with a maximum tree depth of 7 due to a more complex setting. The results are illustrated in Figure 5. REPID correctly identifies only the true interactions for both models. In most of the repetitions, the H-Statistic does not find an interaction between \mathbf{x}_1 and \mathbf{x}_2 for the GAM. A possible reason for this behavior is the rather high main effect of \mathbf{x}_1 compared to the interaction effect (Theorem 4). More experiments of different models and settings – including varying values of λ to obtain shallower or deeper trees – can be found in Appendix B.2. The experiments show that shallow trees produce fewer regions and are therefore easier to interpret. However, they might only detect the most important interactions. Deeper trees are more likely to also identify less important interactions but are less interpretable.

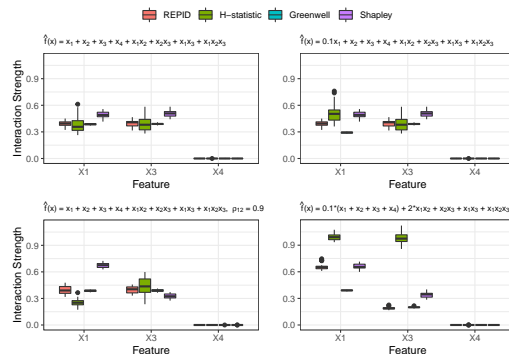


Figure 4: Comparison of REPID, the H-Statistic, Greenwell’s, and SHAP interaction indices for interactions between \mathbf{x}_2 and all other features for 30 repetitions. The upper left plot shows the initial setting (1). The upper and lower right plots adjust effect sizes (2), while the bottom left plot adjusts the correlation (3).

5 REAL-WORLD EXAMPLE

We now demonstrate the usefulness of REPID on the *titanic* data (Dawson, 1995). The labeled part of the dataset consists of 11 characteristics of 891 passengers of the ocean liner Titanic and a binary label if they survived. After some pre-processing steps that are de-

Julia Herbinger, Bernd Bischl, Giuseppe Casalicchio

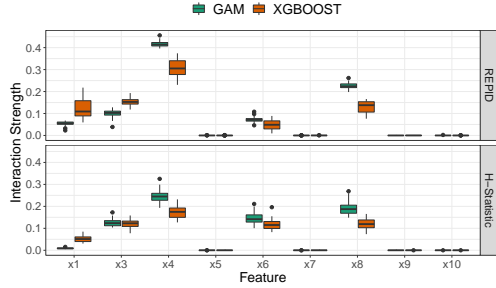


Figure 5: Comparison of the interaction strength between x_2 and all other features measured by REPID (top) and the H-Statistic (bottom) on 30 repetitions. The mean (standard deviation) of the models’ test performance (measured by the mean squared error) is: GAM: 0.36 (0.01), XGBOOST: 0.57 (0.11).

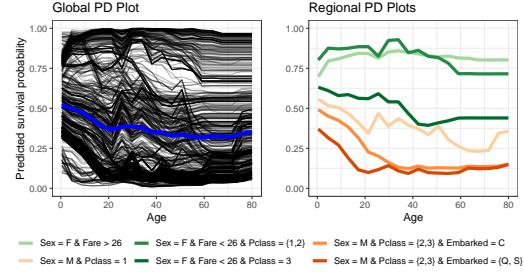


Figure 6: Global PD plot (blue) including ICE curves (left) and the REPs after applying REPID (right) for the feature of interest Age of the *titanic* dataset. The interaction importance $intImp_j$ between Age and the interacting features is 0.28 (Sex), 0.17 (Pclass), 0.13 (Fare), 0.06 (Embarked) and $R_{int}^2 = 0.64$.

scribed in more detail in Appendix B.4, we train a RF with 500 trees on the dataset. Therefore, we obtain a balanced accuracy of 0.8 under 5-fold cross-validation. We are interested in how the age of the passengers affects the probability of survival. The left plot in Figure 6 shows that, from 0 to 20 years, the PD plot for passengers continuously decreases and then flattens above 20 years. The ICE curves indicate that age might influence the predicted survival probability for different passengers in different ways, and thus, interactions with other features might be present. The REPs after applying REPID by using a grid size of 20, a maximum depth of 3, a minimum number of 30 observations, and $\gamma = 0.2$ are shown in the right plot of Figure 6. The 3 most interacting features are Sex, Pclass (passenger class), and Fare. The green REPs show that the predicted survival probability of female passengers is on average higher compared to their male counterparts independent of their age. However, it is also visible that the probability strongly depends on the passenger’s class and the fare they paid. While female passengers who paid a high fare or who belong to an upper or middle class show an overall high survival probability independent of their age (even slightly increasing until 30), the survival probability of women with a low fare and Pclass drops with age. For men from middle and lower classes, the predicted survival probability drops dramatically from 0 until 20 to 30, meaning that for the sub-population of male passengers, the chances of survival are several factors higher for children than for adults.

More real-world examples for the *California housing* (Pace and Barry, 1997) and the *diabetes* (Smith et al., 1988) datasets are provided in Appendix B.4.

6 DISCUSSION

We have introduced the interaction detection method REPID, which provides more representative PD plots on interpretable regions and enables quantification of feature interactions. We have proven its theoretical and empirical advantages and demonstrated how it out-performs alternatives presented in Section 3 and 4. Unlike the H-Statistic or SHAP interaction index, REPID is not influenced by correlations between the feature of interest x_S and other features x_C . However, like the other methods, it might be affected if features within x_C are correlated. Furthermore, the method might be limited if the feature of interest is, e.g., highly skewed, especially if an equidistant grid is used for computations. Possible solutions might be feature transformations or to use a sample or quantile-based grid. As our method is based on a tree-based partitioning algorithm that is known to be unstable (Breiman, 1996), the question arises whether the splitting procedure in Algorithm 1 is a potential limitation. However, with regards to the interaction quantification, we demonstrated in Section 4 that we obtain stable results when repeating the experiments multiple times. A more detailed analysis on the robustness of the method can be found in Appendix B.3.

Author Contributions (CRediT taxonomy)

Contributing authors: Julia Herbinger¹, Bernd Bischl², Giuseppe Casalicchio³. Conceptualization: 1,3; Methodology: 1,3; Project administration: 3; Formal analysis: 1,3; Writing - original draft preparation: 1; Writing - review and editing: 1,2,3; Investigation: 1,3; Visualization: 1; Validation: 1,3; Software: 1,3; Funding acquisition: 2,3; Supervision: 2,3.

Acknowledgements

This work has been partially supported by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A, the Bavarian Ministry of Economic Affairs, Regional Development and Energy as part of the program “Bayerischen Verbundförderprogramms (BayVFP) – Förderlinie Digitalisierung – Förderbereich Informations- und Kommunikationstechnik” under the grant DIK-2106-0007 // DIK0260/02. The authors of this work take full responsibility for its content.

References

- Aas, K., Jullum, M., and Løland, A. (2021). Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *Artificial Intelligence*, 298:103502.
- 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.
- Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.
- Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth.
- Britton, M. (2019). Vine: Visualizing statistical interactions in black box models. *arXiv preprint arXiv:1904.00561*.
- Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, pages 785–794, New York, NY, USA. ACM.
- Dawson, R. J. M. (1995). The “unusual episode” data revisited. *Journal of Statistics Education*, 3(3).
- Fisher, A., Rudin, C., and Dominici, F. (2019). All models are wrong, but many are useful: Learning a variable’s importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177):1–81.
- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232.
- Friedman, J. H., Popescu, B. E., et al. (2008). Predictive learning via rule ensembles. *Annals of Applied Statistics*, 2(3):916–954.
- Fujimoto, K., Kojadinovic, I., and Marichal, J.-L. (2006). Axiomatic characterizations of probabilistic and cardinal-probabilistic interaction indices. *Games and Economic Behavior*, 55(1):72–99.
- Goldstein, A., Kapelner, A., Bleich, J., and Pitkin, E. (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, 24(1):44–65.
- Greenwell, B. M., Boehmke, B. C., and McCarthy, A. J. (2018). A simple and effective model-based variable importance measure. *arXiv preprint arXiv:1805.04755*.
- Grömping, U. (2020). Model-agnostic effects plots for interpreting machine learning models. Report 1/2020, Reports in Mathematics, Physics and Chemistry. Department II, Beuth University of Applied Sciences Berlin.
- Hooker, G. (2004). Discovering additive structure in black box functions. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 575–580. ACM.
- Hooker, G. (2007). Generalized functional anova diagnostics for high-dimensional functions of dependent variables. *Journal of Computational and Graphical Statistics*, 16(3):709–732.
- Hu, L., Chen, J., Nair, V. N., and Sudjianto, A. (2020). Surrogate locally-interpretable models with supervised machine learning algorithms. *arXiv preprint arXiv:2007.14528*.
- Lundberg, S. M., Erion, G. G., and Lee, S.-I. (2018). Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*.
- Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35.
- Molnar, C., Casalicchio, G., and Bischl, B. (2020). Interpretable machine learning—a brief history, state-of-the-art and challenges. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 417–431. Springer.
- Molnar, C., König, G., Bischl, B., and Casalicchio, G. (2021a). Model-agnostic feature importance and effects with dependent features – a conditional subgroup approach. *arXiv preprint arXiv:2006.04628*.
- Molnar, C., König, G., Herbringer, J., Freiesleben, T., Dandl, S., Scholbeck, C. A., Casalicchio, G., Grosse-Wentrup, M., and Bischl, B. (2021b). General pitfalls of model-agnostic interpretation methods for machine learning models. *arXiv preprint arXiv:2007.04131*.
- Pace, R. K. and Barry, R. (1997). Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297.

Julia Herbinger, Bernd Bischl, Giuseppe Casalicchio

- Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). “Why should i trust you?” Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.
- Shapley, L. S. (1953). A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317.
- Smith, J. W., Everhart, J. E., Dickson, W., Knowler, W. C., and Johannes, R. S. (1988). Using the adap learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the annual symposium on computer application in medical care*, page 261. American Medical Informatics Association.
- Strumbelj, E. and Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, 41(3):647–665.
- Su, X., Tsai, C.-L., Wang, H., Nickerson, D. M., and Li, B. (2009). Subgroup analysis via recursive partitioning. *Journal of Machine Learning Research*, 10(2).

Supplementary Material: REPID: Regional Effect Plots with implicit Interaction Detection

A THEORETICAL EVIDENCE

A.1 Proofs

Here, we provide the proofs of the Theorems defined in Section 3. For each Theorem, we first provide a textual description in a proof sketch followed by the formal proof.

Note: For our proofs, we apply the concept of functional decomposition. One concept of functional decomposition has been introduced in Section 2. The so-called functional ANOVA (fANOVA) decomposition is a well-known approach to decompose a function in main and interaction effects. The fANOVA decomposition defined in Section 2 is based on Hooker (2004), and according to this definition, covariates must be independent to obtain a unique decomposition. However, we argue that this is not a relevant issue for our methods, since: (1) We do not try to estimate or calculate the decomposed mean-zero function terms g_W ; we only use the (valid) assumption that a function can be decomposed as in Eq. (3) to prove our theorems. Hence, we are not directly interested in a unique solution of the decomposition. (2) Still, it is possible to relax this assumption by using the generalized fANOVA (Hooker, 2007), which is a weighted version of the “normal” fANOVA to address the extrapolation problem when strong correlations are present. However, it is also possible to use another functional decomposition (e.g., as done in Apley and Zhu (2020)) for these proofs.

A.1.1 Proof of Theorem 1 and Corollary 1.1

Proof Sketch Since $E_{X_C} [\hat{f}(\mathbf{x}_S, \mathbf{x}_C^{(i)})] = \hat{f}(\mathbf{x}_S, \mathbf{x}_C^{(i)})$ and if Eq. (3) holds, the fANOVA decomposition can also be applied to the i -th ICE curve. Since $\mathbf{x}_C^{(i)}$ is constant in i , all fANOVA components that do not depend on \mathbf{x}_S can be summarized to an individual intercept shift of observation i and, thus, cancelled out by mean-centering an ICE curve. The remaining term is then defined by the mean-centered main and mean-centered individual interaction effect of \mathbf{x}_S for observation i . Taking the expected value w.r.t. X_C results in an analogous decomposition of the PD function and mean-centered PD function, respectively.

Proof 1 We first derive the fANOVA decomposition of the i -th ICE curve $\hat{f}(\mathbf{x}_S, \mathbf{x}_C^{(i)})$ using Eq. (3) and use this decomposition to derive the mean-centered version $\hat{f}^c(\mathbf{x}_S, \mathbf{x}_C^{(i)})$ for $|S| = 1$. Therefore, we first decompose the function into main and interaction effects depending on \mathbf{x}_S . *Note:* The term g_0 represents a constant intercept shift. This term is necessary to receive zero-mean functional components, i.e., e.g., $E_X[g_S(X_S)] = 0$.

$$\begin{aligned}
 \hat{f}(\mathbf{x}_S, \mathbf{x}_C^{(i)}) &= E_{X_C|X_C} [\hat{f}(\mathbf{x}_S, X_C)|X_C = \mathbf{x}_C^{(i)}] \\
 &= \underbrace{g_0}_{\text{constant term}} + \underbrace{g_S(\mathbf{x}_S)}_{\text{main effect of } \mathbf{x}_S} + \underbrace{\sum_{j \in C} g_j(\mathbf{x}_j^{(i)})}_{\text{main effect of all other features } \mathbf{x}_j \text{ for observation } i} \\
 &+ \underbrace{\sum_{k=1}^{p-1} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k \cup \{S\}}(\mathbf{x}_S, \mathbf{x}_{C_k}^{(i)})}_{(k+1)\text{-order interaction between } \mathbf{x}_S \text{ and } \mathbf{x}_{C_k} \text{ for observation } i} + \underbrace{\sum_{k=2}^{p-1} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k}(\mathbf{x}_{C_k}^{(i)})}_{k\text{-order interaction between features within } C_k \text{ for observation } i}
 \end{aligned}$$

Julia Herbinger, Bernd Bischl, Giuseppe Casalicchio

$$\begin{aligned}
 \hat{f}^c(\mathbf{x}_S, \mathbf{x}_C^{(i)}) &= \hat{f}(\mathbf{x}_S, \mathbf{x}_C^{(i)}) - E_{X_S} \left[\hat{f}(X_S, \mathbf{x}_C^{(i)}) \right] \\
 &= g_0 + g_S(\mathbf{x}_S) + \sum_{j \in C} g_j(\mathbf{x}_j^{(i)}) + \sum_{k=1}^{p-1} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k \cup \{S\}}(\mathbf{x}_S, \mathbf{x}_{C_k}^{(i)}) + \sum_{k=2}^{p-1} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k}(\mathbf{x}_{C_k}^{(i)}) \\
 &\quad - g_0 - \underbrace{E_{X_S} [g_S(X_S)]}_{=0} - \sum_{j \in C} g_j(\mathbf{x}_j^{(i)}) - E_{X_S} \left[\sum_{k=1}^{p-1} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k \cup \{S\}}(X_S, \mathbf{x}_{C_k}^{(i)}) \right] - \sum_{k=2}^{p-1} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k}(\mathbf{x}_{C_k}^{(i)}) \\
 &= \underbrace{g_S^{cS}(\mathbf{x}_S)}_{\text{mean-centered main effect of } \mathbf{x}_S} + \underbrace{\sum_{k=1}^{p-1} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k \cup \{S\}}(\mathbf{x}_S, \mathbf{x}_{C_k}^{(i)}) - E_{X_S} \left[\sum_{k=1}^{p-1} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k \cup \{S\}}(X_S, \mathbf{x}_{C_k}^{(i)}) \right]}_{\text{mean-centered interaction effect of } \mathbf{x}_S \text{ with } \mathbf{x}_C^{(i)} \text{ for observation } i} \\
 &= \underbrace{g_S^{cS}(\mathbf{x}_S)}_{\text{mean-centered main effect of } \mathbf{x}_S} + \underbrace{\sum_{k=1}^{p-1} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k \cup \{S\}}^{cS}(\mathbf{x}_S, \mathbf{x}_{C_k}^{(i)})}_{\text{mean-centered interaction effect of } \mathbf{x}_S \text{ with } \mathbf{x}_C^{(i)} \text{ for observation } i}
 \end{aligned}$$

Proof 1.1 We first derive the fANOVA decomposition of the PD function $\hat{f}_S^{PD}(\mathbf{x}_S)$ using Eq. (3) and use this decomposition to derive its mean-centered version $f_S^{PD,c}(\mathbf{x}_S)$ for $|S| = 1$.

$$\begin{aligned}
 f_S^{PD}(\mathbf{x}_S) &= E_{X_C} \left[\hat{f}(\mathbf{x}_S, X_C) \right] \\
 &= E_{X_C} \left[g_0 + g_S(\mathbf{x}_S) + \sum_{j \in C} g_j(\mathbf{x}_j^{(i)}) + \sum_{k=1}^{p-1} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k \cup \{S\}}(\mathbf{x}_S, X_{C_k}) + \sum_{k=2}^{p-1} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k}(X_{C_k}) \right] \\
 &= g_0 + g_S(\mathbf{x}_S) + \underbrace{E_{X_C} \left[\sum_{j \in C} g_j(\mathbf{x}_j^{(i)}) \right]}_{\text{expected main effect of features in } \mathbf{x}_C (=0)} + \underbrace{E_{X_C} \left[\sum_{k=1}^{p-1} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k \cup \{S\}}(\mathbf{x}_S, X_{C_k}) \right]}_{\text{expected interaction effect of features in } \mathbf{x}_C (=0)} + \underbrace{E_{X_C} \left[\sum_{k=2}^{p-1} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k}(X_{C_k}) \right]}_{\text{expected interaction effect of features in } \mathbf{x}_C (=0)} \\
 &= g_0 + \underbrace{g_S(\mathbf{x}_S)}_{\text{main effect of } \mathbf{x}_S} + \underbrace{E_{X_C} \left[\sum_{k=1}^{p-1} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k \cup \{S\}}(\mathbf{x}_S, X_{C_k}) \right]}_{\text{expected interaction effect of } \mathbf{x}_S \text{ with } \mathbf{x}_C \text{ w.r.t. } \mathbf{x}_C}
 \end{aligned}$$

If the expected value of each decomposed term $g(\mathbf{x})$ exists and if the integral of the absolute value is finite, then Fubini's theorem can be applied, and the mean-centered PD function of \mathbf{x}_S for $|S| = 1$ can be derived by:

$$\begin{aligned}
 f_S^{PD,c}(\mathbf{x}_S) &= f_S^{PD}(\mathbf{x}_S) - E_{X_S} [f_S^{PD}(X_S)] \\
 &= E_{X_C} \left[\hat{f}(\mathbf{x}_S, X_C) \right] - E_{X_S} \left[g_0 + g_S(X_S) + E_{X_C} \left[\sum_{k=1}^{p-1} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k \cup \{S\}}(X_S, X_{C_k}) \right] \right]
 \end{aligned}$$

$$\begin{aligned}
 &= g_0 + g_S(\mathbf{x}_S) + E_{X_C} \left[\sum_{k=1}^{p-1} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k \cup \{S\}}(\mathbf{x}_S, X_{C_k}) \right] \\
 &- g_0 - \underbrace{E_{X_S} [g_S(X_S)]}_{=0} - \underbrace{E_{X_S} \left[E_{X_C} \left[\sum_{k=1}^{p-1} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k \cup \{S\}}(X_S, X_{C_k}) \right] \right]}_{\text{expected interaction effect between } \mathbf{x}_S \text{ and } \mathbf{x}_C (=0)} \\
 &= \underbrace{g_S^{cS}(\mathbf{x}_S)}_{\text{mean-centered main effect of } \mathbf{x}_S} + E_{X_C} \left[\sum_{k=1}^{p-1} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k \cup \{S\}}^{cS}(\mathbf{x}_S, X_{C_k}) \right] \\
 &\quad \underbrace{\hspace{10em}}_{\text{expected mean-centered interaction effect of } \mathbf{x}_S \text{ with } \mathbf{x}_C \text{ w.r.t. } \mathbf{x}_C}
 \end{aligned}$$

A.1.2 Proof of Theorem 2

Proof Sketch If the function $\hat{f}(\mathbf{x})$ can be decomposed as in Eq. (3), then Theorem 1 and Corollary 1.1 hold, and the main effect of \mathbf{x}_S is cancelled out when calculating $\mathcal{R}_{L2}(\mathcal{N}_g)$. The remaining term is given by the distance between the i -th centered interaction effect and the average centered interaction effect between \mathbf{x}_S and \mathbf{x}_C .

Proof 2 In the risk function of Eq. (6), the squared distance between the i -th mean-centered ICE curve $\hat{f}^c(\mathbf{x}_S, \mathbf{x}_C^{(i)})$ and the respective PD function $f_S^{PD,c}(\mathbf{x}_S)$ is calculated. The distance can be reduced to the following term:

$$\begin{aligned}
 \hat{f}^c(\mathbf{x}_S, \mathbf{x}_C^{(i)}) - f_S^{PD,c}(\mathbf{x}_S) &= g_S^{cS}(\mathbf{x}_S) + \sum_{k=1}^{p-1} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k \cup \{S\}}^{cS}(\mathbf{x}_S, \mathbf{x}_C^{(i)}) - g_S^{cS}(\mathbf{x}_S) - \sum_{k=1}^{p-1} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} E_{X_C} [g_{C_k \cup \{S\}}^{cS}(\mathbf{x}_S, X_{C_k})] \\
 &= \sum_{k=1}^{p-1} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} (g_{C_k \cup \{S\}}^{cS}(\mathbf{x}_S, \mathbf{x}_C^{(i)}) - E_{X_C} [g_{C_k \cup \{S\}}^{cS}(\mathbf{x}_S, X_{C_k})])
 \end{aligned}$$

The first term is the mean-centered interaction effect of the i -th ICE curve, while the second term represents the mean-centered expected interaction effect over the joint distribution of \mathbf{x}_C (which is included in the mean-centered PD function, see also the decomposition of the mean-centered PD function $f_S^{PD,c}(\mathbf{x}_S)$ in the proof in Appendix A.1.1). The intuition behind our split criterion is that we search for the optimal split value of a feature in \mathbf{x}_C that reduces the aggregated variance over all curves the most if we split according to this optimal split value. Thus, we try to find regions in the feature space \mathbf{x}_C where the distance between the individual centered ICE curves in this region and the respective mean-centered PD plot is as small as possible. Hence, we want to minimize the deviation of the individual interaction effect of the ICE curves in a region from the average interaction effect in the considered region.

A.1.3 Proof of Theorem 4

Proof Sketch The two-way interaction index of the H-Statistic is calculated by dividing the variance of the difference between the centered 2-dimensional PD plot and the 1-dimensional PD plots of the two features of interest (nominator) by the variance of the centered 2-dimensional PD plot (denominator, see Eq. (2)). If Eq. (3) holds, we can apply Theorem 1 and Corollary 1.1, and it can be shown that the main effects of the two features of interest are cancelled out in the nominator, but are still present in the denominator (scaling factor) of the interaction index.

Julia Herbinger, Bernd Bischl, Giuseppe Casalicchio

Proof 4 Let $S = \{j, l\}$ and $C = S^c$ its complement, then the 2-dimensional PD function $f_S^{PD}(\mathbf{x}_S)$ of \mathbf{x}_j and \mathbf{x}_l is given by

$$\begin{aligned} f_S^{PD}(\mathbf{x}_S) &= E_{X_C} [f(\mathbf{x}_S, X_C)] \\ &= g_0 + g_j(\mathbf{x}_j) + g_l(\mathbf{x}_l) + g_{jl}(\mathbf{x}_j, \mathbf{x}_l) + E_{X_C} \left[\underbrace{\sum_{k=1}^{p-2} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k}(X_{C_k})}_{\text{expected interaction effect of features in } \mathbf{x}_C (=0)} \right] \\ &\quad + E_{X_C} \left[\sum_{k=1}^{p-2} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k \cup \{j\}}(\mathbf{x}_j, X_{C_k}) + g_{C_k \cup \{l\}}(\mathbf{x}_l, X_{C_k}) + g_{C_k \cup \{S\}}(\mathbf{x}_S, X_{C_k}) \right] \end{aligned}$$

If the expected value of each decomposed term $g(\mathbf{x})$ exists, and if the integral of the absolute value is finite, then Fubini's theorem can be applied, and the mean-centred 2-dimensional PD function $f_S^{PD,c}(\mathbf{x}_S)$ of features \mathbf{x}_j and \mathbf{x}_l can then be derived by

$$\begin{aligned} f_S^{PD,c}(\mathbf{x}_S) &= f_S^{PD}(\mathbf{x}_S) - E_{X_S} [f_S^{PD,c}(X_S)] \\ &= g_0 + g_j(\mathbf{x}_j) + g_l(\mathbf{x}_l) + g_{jl}(\mathbf{x}_j, \mathbf{x}_l) \\ &\quad + E_{X_C} \left[\sum_{k=1}^{p-2} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k \cup \{j\}}(\mathbf{x}_j, X_{C_k}) + g_{C_k \cup \{l\}}(\mathbf{x}_l, X_{C_k}) + g_{C_k \cup \{S\}}(\mathbf{x}_S, X_{C_k}) \right] \\ &\quad - g_0 - \underbrace{E_{X_S} [g_j(X_j) + g_l(X_l) + g_{jl}(X_j, X_l)]}_{=0} \\ &\quad - E_{X_S} \left[\underbrace{E_{X_C} \left[\sum_{k=1}^{p-2} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k \cup \{j\}}(X_j, X_{C_k}) + g_{C_k \cup \{l\}}(X_l, X_{C_k}) + g_{C_k \cup \{S\}}(X_S, X_{C_k}) \right]}_{\text{expected interaction effect between } \mathbf{x}_S \text{ and } \mathbf{x}_C (=0)} \right] \\ &= \underbrace{g_j^{cS}(\mathbf{x}_j) + g_l^{cS}(\mathbf{x}_l)}_{\text{mean-centered main effects of } \mathbf{x}_S} + \underbrace{g_{jl}^{cS}(\mathbf{x}_j, \mathbf{x}_l)}_{\text{mean-centered interaction effect between } \mathbf{x}_j \text{ and } \mathbf{x}_l} \\ &\quad + E_{X_C} \left[\sum_{k=1}^{p-2} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k \cup \{j\}}^{cS}(\mathbf{x}_j, X_{C_k}) + g_{C_k \cup \{l\}}^{cS}(\mathbf{x}_l, X_{C_k}) + g_{C_k \cup \{S\}}^{cS}(\mathbf{x}_S, X_{C_k}) \right] \\ &\quad \underbrace{\hspace{10em}}_{\text{expected mean-centered interaction effects between features in } \mathbf{x}_S \text{ and features in } \mathbf{x}_C \text{ w.r.t. } \mathbf{x}_C} \end{aligned}$$

It follows that the H-Statistic still depends on the mean-centered main effects $g_j^{cS}(\mathbf{x}_j)$ and $g_l^{cS}(\mathbf{x}_l)$ of \mathbf{x}_j and \mathbf{x}_l in the denominator.

To calculate the nominator of the H-Statistic, we must subtract the 1-dimensional mean-centered PD functions of x_j and x_l as follows:

$$f_S^{PD,c}(\mathbf{x}_S) - f_j^{PD,c}(\mathbf{x}_j) - f_l^{PD,c}(\mathbf{x}_l) = g_j^{cS}(\mathbf{x}_j) + g_l^{cS}(\mathbf{x}_l) + g_{jl}^{cS}(\mathbf{x}_j, \mathbf{x}_l)$$

$$\begin{aligned}
 & + E_{X_C} \left[\sum_{k=1}^{p-2} \sum_{\substack{C_k \subseteq C, \\ |C_k|=k}} g_{C_k \cup \{j\}}^{c_S}(\mathbf{x}_j, X_{C_k}) + g_{C_k \cup \{l\}}^{c_S}(\mathbf{x}_l, X_{C_k}) + g_{C_k \cup \{S\}}^{c_S}(\mathbf{x}_S, X_{C_k}) \right] \\
 & - g_j^{c_S}(\mathbf{x}_j) - \sum_{k=1}^{p-1} \sum_{\substack{C_k \subseteq C \cup \{l\}, \\ |C_k|=k}} E_{X_{C \cup \{l\}}} \left[g_{C_k \cup \{j\}}^{c_S}(\mathbf{x}_j, X_{C_k}) \right] \\
 & - g_l^{c_S}(\mathbf{x}_l) - \sum_{k=1}^{p-1} \sum_{\substack{C_k \subseteq C \cup \{j\}, \\ |C_k|=k}} E_{X_{C \cup \{j\}}} \left[g_{C_k \cup \{l\}}^{c_S}(\mathbf{x}_l, X_{C_k}) \right]
 \end{aligned}$$

Thus, in the nominator of the H-Statistic, the variance of the calculated term is determined. This term only depends on interactions with features \mathbf{x}_j and \mathbf{x}_l , while the main effects $g_j^{c_S}(\mathbf{x}_j)$ and $g_l^{c_S}(\mathbf{x}_l)$ that are present in the denominator are cancelled out.

A.1.4 Proof of Theorem 5

Proof Sketch The loss function in Eq. (5), which is used for the splitting in Algorithm 1, is calculated grid-wise. This means that we calculate the variation measured by the estimated variance (L2 loss) for each grid point $x_S^{(k)}$ with $k \in \{1, \dots, m\}$. Hence, \mathbf{x}_S is not treated as a random variable but as a constant. It follows that when calculating the variance over all ICE curves on a specific grid point $x_S^{(k)}$, no covariance terms between X_S and features in X_C are considered.

Proof 5 $\mathcal{L}(\mathcal{N}_g, x_S)$ of Eq. (5) is estimated by taking the variance over all mean-centered ICE curves within a region \mathcal{N}_g for a fixed grid point of \mathbf{x}_S . Hence, for each grid point $k \in \{1, \dots, m\}$, we calculate:

$$\mathcal{L}(x_S^{(k)}, \mathcal{N}_g) = \text{Var}_{X|\mathcal{N}_g}(\hat{f}^c(X)|X_S = x_S^{(k)}) = \text{Var}_{X|\mathcal{N}_g}[\hat{f}^c(x_S^{(k)}, X_C)].$$

Since $x_S^{(k)}$ is constant, it follows $\text{Var}_{X|\mathcal{N}_g}[\hat{f}^c(x_S^{(k)}, X_C)] = \text{Var}_{X_C|\mathcal{N}_g}[\hat{f}^c(x_S^{(k)}, X_C)]$, and hence, the calculated variance only depends on features in C while there are no covariance terms between X_S and features in X_C included.

A.2 Derivation of R Squared Measure

Let $d = 0, \dots, D$ be the depth of the tree, where $d = 0$ is the depth of the root node and $d = D$ of the leaf nodes of a symmetric tree, and k defines the index of the node at each depth from left to right (starting from 0). With a slight abuse of notation, we denote \mathcal{R}_k^d as the risk of the k -th node at depth d . For example, \mathcal{R}_0^0 is the risk of the root node ($\mathcal{R}(\mathcal{N})$). Let $\mathcal{B}_t = \mathcal{B}_P^0$ denote the subset of terminal nodes in a symmetric tree. We can derive an interaction-related R^2 measure by aggregating the interaction importance over all parent nodes \mathcal{B}_P :

$$\begin{aligned}
 R_{int}^2 &= \sum_{P \in \mathcal{B}_P} \text{intImp}(\mathcal{N}_P) \\
 &= \frac{1}{\mathcal{R}_0^0} \cdot \sum_{d=0}^{D-1} \sum_{k=0}^d (\mathcal{R}_k^d - \mathcal{R}_{2k}^{d+1} - \mathcal{R}_{2k+1}^{d+1}) \\
 &= \frac{1}{\mathcal{R}_0^0} \cdot (\mathcal{R}_0^0 - \sum_{k=0}^{D-1} (\mathcal{R}_{2k}^D + \mathcal{R}_{2k+1}^D)) \\
 &= 1 - \frac{\sum_{k=0}^{D-1} (\mathcal{R}_{2k}^D + \mathcal{R}_{2k+1}^D)}{\mathcal{R}_0^0} \\
 &= 1 - \frac{\sum_{t \in \mathcal{B}_t} \mathcal{R}(\mathcal{N}_t)}{\mathcal{R}(\mathcal{N})}
 \end{aligned}$$

Julia Herbinger, Bernd Bischl, Giuseppe Casalicchio

Explanation: According to Eq. (7), $intImp(\mathcal{N}_P)$ is defined by $intImp(\mathcal{N}_P) = \frac{\mathcal{R}(\mathcal{N}_P) - (\mathcal{R}(\mathcal{N}_l) + \mathcal{R}(\mathcal{N}_r))}{\mathcal{R}(\mathcal{N})}$ which is, e.g., for the first split (using the new notation defined in this section) the same as $intImp(\mathcal{N}) = \frac{\mathcal{R}_0^0 - (\mathcal{R}_0^1 + \mathcal{R}_1^1)}{\mathcal{R}_0^0}$ and for the split of the first left and right child nodes (which we denote here by \mathcal{N}_l and \mathcal{N}_r , respectively), we obtain $intImp(\mathcal{N}_l) = \frac{\mathcal{R}_0^1 - (\mathcal{R}_0^2 + \mathcal{R}_1^2)}{\mathcal{R}_0^0}$ and $intImp(\mathcal{N}_r) = \frac{\mathcal{R}_1^1 - (\mathcal{R}_2^2 + \mathcal{R}_3^2)}{\mathcal{R}_0^0}$. It follows that, after the second split ($D = 2$), R_{int}^2 can be calculated by

$$\begin{aligned}
 R_{int}^2 &= intImp(\mathcal{N}) + intImp(\mathcal{N}_l) + intImp(\mathcal{N}_r) \\
 &= \frac{1}{\mathcal{R}_0^0} (\mathcal{R}_0^0 - (\mathcal{R}_0^1 + \mathcal{R}_1^1) + \mathcal{R}_0^1 - (\mathcal{R}_0^2 + \mathcal{R}_1^2) + \mathcal{R}_1^1 - (\mathcal{R}_2^2 + \mathcal{R}_3^2)) \\
 &= \frac{1}{\mathcal{R}_0^0} \cdot \sum_{d=0}^1 \sum_{k=0}^d (\mathcal{R}_k^d - \mathcal{R}_{2k}^{d+1} - \mathcal{R}_{2k+1}^{d+1}) \\
 &= \frac{1}{\mathcal{R}_0^0} (\mathcal{R}_0^0 - (\mathcal{R}_0^2 + \mathcal{R}_1^2)) - (\mathcal{R}_2^2 + \mathcal{R}_3^2) \\
 &= \frac{1}{\mathcal{R}_0^0} \cdot (\mathcal{R}_0^0 - \sum_{k=0}^1 (\mathcal{R}_{2k}^{D=2} + \mathcal{R}_{2k+1}^{D=2})) \\
 &= 1 - \frac{\sum_{k=0}^{D-1} (\mathcal{R}_{2k}^D + \mathcal{R}_{2k+1}^D)}{\mathcal{R}_0^0} \\
 &= 1 - \frac{\sum_{t \in \mathcal{B}_t} \mathcal{R}(\mathcal{N}_t)}{\mathcal{R}(\mathcal{N})}
 \end{aligned}$$

From the second to the fourth line of the equation, we can see that the parent nodes (besides the root node) are cancelled out when aggregating the interaction importance over all nodes. It follows that only the deviation between the root node risk and the sum over all terminal node risks remains in the nominator. The denominator is always the root node (baseline) risk.

A.3 Explanations for Weaknesses of other Methods

A.3.1 Small Main Effects

For REPID, we proved with Theorem 2 that the split criterion only depends on interaction effects with the feature of interest \mathbf{x}_S and is independent of main effects. On the other hand, according to Theorem 4, the H-Statistic depends on main effects in the denominator of the H-Statistic. Since the main effect of feature \mathbf{x}_1 is reduced from 1 to 0.1 in the adjusted example of Section 4.1, the denominator of H-Statistic decreases, and hence, the overall H-Statistic value increases for feature \mathbf{x}_1 .

Since we provided proofs for REPID and for the H-Statistic, we will not go into more detail here, but instead derive explanations for the SHAP and Greenwell's interaction indices with regards to varying main effects.

SHAP interaction index By definition, SHAP interaction values only contain the interaction effect between the two features of interest and do not contain their main effects. Since we only sum up the absolute interaction values and divide them by the total amount of two-way interaction values between the feature of interest and all other features, there are also no main effects included in the global SHAP interaction index. Hence, varying main effects does not change the interaction strength / ranking calculated by the SHAP interaction index.

Example: Due to the complexity of an increasing number of feature permutations, we show this relationship on the following simple model: $\hat{f}(\mathbf{x}) = \hat{\beta}_1 \mathbf{x}_1 + \hat{\beta}_2 \mathbf{x}_2 + \hat{\beta}_{12} \mathbf{x}_1 \mathbf{x}_2$ with $E(X_1) = E(X_2) = 0$.

In this case, we can straightforwardly calculate the individual components of the SHAP interaction value with $S = \emptyset$:

$$f_{S \cup \{1,2\}}^{PD}(\mathbf{x}_{S \cup \{1,2\}}) = \hat{\beta}_1 \mathbf{x}_1 + \hat{\beta}_2 \mathbf{x}_2 + \hat{\beta}_{12} \mathbf{x}_1 \mathbf{x}_2$$

Since $E(X_1) = E(X_2) = 0$, it follows:

$$f_{S \cup \{1\}}^{PD}(\mathbf{x}_{S \cup \{1\}}) = \hat{\beta}_1 \mathbf{x}_1 \text{ and } f_{S \cup \{2\}}^{PD}(\mathbf{x}_{S \cup \{2\}}) = \hat{\beta}_2 \mathbf{x}_2 \text{ and } f_S^{PD}(\mathbf{x}_S) = E_X [\hat{f}(X)] = \hat{\beta}_{12} E_X [X_1 X_2]$$

and hence, the SHAP interaction value between \mathbf{x}_1 and \mathbf{x}_2 is given by

$$\begin{aligned}\Phi_{1,2}(\mathbf{x}) &= \frac{1}{2}(f_{S \cup \{1,2\}}^{PD} - f_{S \cup \{1\}}^{PD}(\mathbf{x}_{S \cup \{1\}}) - f_{S \cup \{2\}}^{PD}(\mathbf{x}_{S \cup \{2\}}) + f_S^{PD}(\mathbf{x}_S)) \\ &= \frac{1}{2}(\hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_{12} x_1 x_2 - \hat{\beta}_1 x_1 - \hat{\beta}_2 x_2 + \hat{\beta}_{12} E_X [X_1 X_2]) \\ &= \frac{1}{2}(\hat{\beta}_{12} x_1 x_2 + \hat{\beta}_{12} E_X [X_1 X_2])\end{aligned}$$

Greenwell's interaction index Greenwell et al. (2018) defines feature importance $i(\mathbf{x}_j)$ as the standard deviation over the PD function of a feature \mathbf{x}_j with m_j unique values as follows:

$$i(\mathbf{x}_j)^2 = \frac{1}{m_j - 1} \sum_{k=1}^{m_j} \left(\hat{f}_j^{PD}(x_j^{(k)}) - \frac{1}{m_j} \sum_{k=1}^{m_j} \hat{f}_j^{PD}(x_j^{(k)}) \right)^2$$

To calculate the interaction between \mathbf{x}_j and \mathbf{x}_l , they define the conditional importance $i(\mathbf{x}_j | \mathbf{x}_l = \mathbf{x}_l^{(i)})$ of a feature \mathbf{x}_j given the t -th unique value of \mathbf{x}_l as follows:

$$i(\mathbf{x}_j | \mathbf{x}_l = x_l^{(t)})^2 = \frac{1}{m_j - 1} \sum_{k=1}^{m_j} \left(\hat{f}_j^{PD}(x_j^{(k)} | \mathbf{x}_l = x_l^{(t)}) - \frac{1}{m_j} \sum_{k=1}^{m_j} \hat{f}_j^{PD}(x_j^{(k)} | \mathbf{x}_l = x_l^{(t)}) \right)^2$$

With m_j and m_l being the number of unique values of \mathbf{x}_j and \mathbf{x}_l , respectively, the interaction measure $i(\mathbf{x}_j, \mathbf{x}_l)$ between these two features is then defined by:

$$\begin{aligned}i(\mathbf{x}_j, \mathbf{x}_l) &= \frac{1}{2} \sqrt{\frac{1}{m_l - 1} \sum_{t=1}^{m_l} \left[i(\mathbf{x}_j | \mathbf{x}_l = x_l^{(t)}) - \frac{1}{m_l} \sum_{t=1}^{m_l} i(\mathbf{x}_j | \mathbf{x}_l = x_l^{(t)}) \right]^2} \\ &\quad + \frac{1}{2} \sqrt{\frac{1}{m_j - 1} \sum_{k=1}^{m_j} \left[i(\mathbf{x}_l | \mathbf{x}_j = x_j^{(k)}) - \frac{1}{m_j} \sum_{k=1}^{m_j} i(\mathbf{x}_l | \mathbf{x}_j = x_j^{(k)}) \right]^2}\end{aligned}$$

Instead of conditioning on all features in C as done for ICE curves, Greenwell et al. (2018) conditions only on the second feature of interest (e.g., \mathbf{x}_l) to calculate the variation of PD curves for the first feature of interest (e.g., \mathbf{x}_j). Hence, they first take the variation of each conditioned curve and then calculate the variation over all these curves. Since they calculate the squared distance of each conditioned PD curve to its mean, the distance still contains the main effects of the two features of interest (see Theorem 1).

A.3.2 Dependencies between the Feature of Interest and other Features

For REPID, we proved with Theorem 5 that the loss function of Eq. (5) (which is used for splitting) is not affected by dependencies between the feature of interest \mathbf{x}_S and features in \mathbf{x}_C .

Hence, we will now derive explanations for the H-Statistic, the SHAP, and the Greenwell's interaction indices with regards to dependencies between the feature of interest and other features.

The H-Statistic The H-Statistic (which is estimated as in Eq. (2)) divides the variance of the difference between the mean-centered 2-dimensional PD plot and the two mean-centered 1-dimensional PD plots by the variance of the mean-centered 2-dimensional PD plot. Both the nominator and the denominator depend on the joint distribution of the two features of interest and, hence, also on the dependency between the two features.

Example Considering our simulation example in Section 4.1 with $E(X_1) = E(X_2) = E(X_3) = E(X_4) = 0$, the mean-centered 2-dimensional PD function between \mathbf{x}_1 and \mathbf{x}_2 with $S = \{1, 2\}$ is given by:

$$\hat{f}_S^{PD,c}(\mathbf{x}_1, \mathbf{x}_2) = \hat{\beta}_1 \mathbf{x}_1 + \hat{\beta}_2 \mathbf{x}_2 + \hat{\beta}_3 E(X_3) + \hat{\beta}_{12} \mathbf{x}_1 \mathbf{x}_2 + \hat{\beta}_{23} E(X_3) \mathbf{x}_2 + \hat{\beta}_{13} \mathbf{x}_1 E(X_3) + \hat{\beta}_{123} \mathbf{x}_1 E(X_3) \mathbf{x}_2$$

Julia Herbringer, Bernd Bischl, Giuseppe Casalicchio

$$\begin{aligned}
 & -\hat{\beta}_1 E(X_1) - \hat{\beta}_2 E(X_2) - \hat{\beta}_3 E(X_3) - \hat{\beta}_{12} E_{X_S} [X_1 X_2] - \hat{\beta}_{23} E(X_3) E(X_2) - \hat{\beta}_{13} E(X_1) E(X_3) \\
 & - \hat{\beta}_{123} E_{X_S} [X_1 X_2] E(X_3) \\
 & = \hat{\beta}_1 \mathbf{x}_1 + \hat{\beta}_2 \mathbf{x}_2 + \hat{\beta}_{12} (\mathbf{x}_1 \mathbf{x}_2 - E_{X_S} [X_1 X_2])
 \end{aligned}$$

Calculating the denominator by taking the variance

$$\begin{aligned}
 Var(\hat{f}_S^{PD,c}(\mathbf{x}_1, \mathbf{x}_2)) &= E \left[(\hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 + \hat{\beta}_{12} (X_1 X_2 - E_{X_S} [X_1 X_2]))^2 \right] \\
 & - E \left[\hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 + \hat{\beta}_{12} (X_1 X_2 - E_{X_S} [X_1 X_2]) \right]^2 \\
 &= E \left[\hat{\beta}_1^2 X_1^2 + 2\hat{\beta}_1 \hat{\beta}_2 X_1 X_2 + \hat{\beta}_2^2 X_2^2 + 2\hat{\beta}_1 \hat{\beta}_{12} X_1^2 X_2 + 2\hat{\beta}_2 \hat{\beta}_{12} X_1 X_2^2 \right] \\
 & + E \left[-2\hat{\beta}_1 \hat{\beta}_{12} X_1 E_{X_S} [X_1 X_2] - 2\hat{\beta}_2 \hat{\beta}_{12} X_2 E_{X_S} [X_1 X_2] + \hat{\beta}_{12}^2 X_1^2 X_2^2 \right] \\
 & + E \left[-2\hat{\beta}_{12}^2 X_1 X_2 E_{X_S} [X_1 X_2] + \hat{\beta}_{12}^2 E_{X_S} [X_1 X_2]^2 \right] \\
 &= \hat{\beta}_1^2 Var(X_1) + \hat{\beta}_2^2 Var(X_2) + \hat{\beta}_{12}^2 Var(X_1 X_2) \\
 & + 2\hat{\beta}_1 \hat{\beta}_2 Cov(X_1, X_2) + 2\hat{\beta}_1 \hat{\beta}_{12} Cov(X_1^2, X_2) + 2\hat{\beta}_2 \hat{\beta}_{12} Cov(X_1, X_2^2)
 \end{aligned}$$

in the nominator, we subtract the mean-centered 1-dimensional PD functions (i.e., $\hat{f}_1^{PD,c}(\mathbf{x}_1) = \hat{\beta}_1 \mathbf{x}_1$ and $\hat{f}_2^{PD,c}(\mathbf{x}_2) = \hat{\beta}_2 \mathbf{x}_2$) and take the variance, which results in

$$\begin{aligned}
 & E_X \left[\hat{\beta}_{12}^2 (X_1 X_2 - E_{X_S} [X_1 X_2])^2 \right] - E_X \left[\hat{\beta}_{12} (X_1 X_2 - E_{X_S} [X_1 X_2]) \right]^2 \\
 &= E_X \left[\hat{\beta}_{12}^2 X_1^2 X_2^2 - 2\hat{\beta}_{12}^2 X_1 X_2 E_{X_S} [X_1 X_2] + \hat{\beta}_{12}^2 E_{X_S} [X_1 X_2]^2 \right] \\
 &= \hat{\beta}_{12}^2 Var(X_1 X_2) \\
 &= \hat{\beta}_{12}^2 (Var(X_1)Var(X_2)) - Cov(X_1, X_2)^2 + Cov(X_1^2, X_2^2)
 \end{aligned}$$

It follows that by increasing the correlation between \mathbf{x}_1 and \mathbf{x}_2 to $\rho_{12} = 0.9$, the denominator of the H-Statistic increases compared to the nominator for the given example, and hence, the H-Statistic value between \mathbf{x}_1 and \mathbf{x}_2 decreases compared to the H-Statistic value between \mathbf{x}_2 and \mathbf{x}_3 .

Some general rules that were applied here:

- 1 Rearrangement of variance formula for functions: $Var(g(X)) = E[g(X)^2] - (E[g(X)])^2$
- 2 Expected value of a product of two random variables: $E[X_1 X_2] = E[X_1]E[X_2] + Cov(X_1, X_2)$ which reduces for $E(X_1) = E(X_2) = 0$ to $E[X_1 X_2] = Cov(X_1, X_2)$
- 3 Variance of a product of two random variables: $V(XY) = E[X^2 Y^2] - (E[XY])^2 = Cov(X^2, Y^2) + (V(X) + (E[X]^2)(V(Y) + (E[Y]^2) - (Cov(X, Y) + E[X]E[Y])^2$ which reduces for $E[X] = E[Y] = 0$ to $V(XY) = Cov(X^2, Y^2) + V(X)V(Y) - Cov(X, Y)^2$

SHAP interaction index SHAP interaction values – and with that, also the (global) SHAP Interaction index – depend on the correlation between the two features of interest, since we consider the joint distribution of the features as we do for the H-Statistic.

Example In Appendix A.3.1, we derived the SHAP interaction value for a simple linear model of two features with a positive linear interaction between these features, which resulted in

$$\Phi_{1,2}(\mathbf{x}) = \frac{1}{2}(\hat{\beta}_{12} \mathbf{x}_1 \mathbf{x}_2 + \hat{\beta}_{12} E_X [X_1 X_2])$$

Hence, if \mathbf{x}_1 and \mathbf{x}_2 are positively correlated as in our example in Section 4.1, then $E_X [X_1 X_2] > 0$, while this term is 0 if the two features are independent. This is why \mathbf{x}_1 shows a higher interaction value than \mathbf{x}_3 in the referred simulation study.

Greenwell’s interaction index Similarly to our approach, the Greenwell’s interaction index conditions on one of the two features of interest. They calculate the variance w.r.t. the other feature of interest, and vice versa. Hence, the dependency between the two regarded features does not influence the resulting interaction index.

B EMPIRICAL EVIDENCE

In this section, we provide more empirical evidence for the usefulness of REPID. We will further analyze the nonlinear simulation setting described in Section 4.2 and will also look at a linear example where interactions can clearly be ranked. Furthermore, we analyze the influence of the improvement parameter γ used as stop criterion and provide some evidence for the robustness of our method in Section B.3. In Section B.4, we clarify the pre-processing steps of the real-world example that was analyzed in Section 5.

Infrastructure All experiments only require CPUs (and no GPUs) and were computed on a Linux cluster (see Table 2).

Table 2: Description of the infrastructure used for the experiments in this paper.

Computing Infrastructure	
Type	Linux CPU Cluster
Architecture	28-way Haswell-EP nodes
Cores per Node	1
Memory limit (per core)	2.2 GB

B.1 Overview on Weaknesses of other Methods

In Table 3, we provide a brief overview of the simulation setting, including a sensitivity analysis that we performed in Section 4.1. The table shows that only REPID provides on average correct ranks for all settings, while the other state-of-the-art methods provide for at least one of the settings a wrong ranking (on average).

Table 3: Summary table of settings and key results of the simulation study in Section 4.1. The column “Setting” refers to the setting number in Section 4.1. The second column refers to the adjustments made in the setting compared to the initial setting. The other four columns show if the average ranks (r) of the feature interactions with the feature of interest (\mathbf{x}_2) are correct (meaning that the ranks are the same as the ranks of the underlying data-generating process and fitted linear model) or if they are wrong (different from the ranks in the data-generating process and fitted linear model).

Setting	Adjustment	REPID	H-Statistic	Greenwell	Shapley
(2)	$\beta_1 = 0.1$ (initial: 1)	correct $r(x_1) = r(x_3)$	wrong $r(x_1) > r(x_3)$	wrong $r(x_1) < r(x_3)$	correct
(2)	$\beta_1 = \beta_2 = \beta_3 = \beta_4 = 0.1$ and $\beta_{12} = 2$ (initial: 1)	correct $r(x_1) > r(x_3)$	wrong $r(x_1) = r(x_3)$	correct	correct
(3)	$\rho_{12} = 0.9$ (initial: 0)	correct $r(x_1) = r(x_3)$	wrong $r(x_1) < r(x_3)$	correct	wrong $r(x_1) > r(x_3)$

B.2 Further experiments

Nonlinear example In Section 4.2, we compared REPID and the H-Statistic for the interactions between the most interacting feature \mathbf{x}_2 and the other nine features of the simulation setting described in the referred section. In addition to the correctly specified GAM and XGBOOST model from Section 4.2, we now also compare the results to two other ML models: an RF with 500 trees – the mean and standard deviation of the models’ test performance (measured by the mean squared error) is 1.01 and 0.16 – and a support vector machine (SVM) using epsilon support vector regression with a Gauss kernel, $C = 1$ and $\epsilon = 0.1$ – the mean and standard deviation of the models’ test performance (measured by the mean squared error) is 0.76 and 0.07. The left plot in Figure 7 shows the same illustration as in Figure 5 for the interactions between the non-influential feature \mathbf{x}_{10} and all

Julia Herbinger, Bernd Bischl, Giuseppe Casalicchio

other features. For the correctly specified GAM and XGBOOST model, both methods do – as expected – on average not find any interactions. While REPID on average also recognizes that there are no interactions present between x_{10} and all other features for the SVM and RF models, the H-Statistic finds some higher interactions, especially for the SVM. A possible explanation is that x_{10} does also not influence the target by a main effect in the underlying function, and hence, possible small found interaction effects might lead to high H-Statistic values. The outliers for some features when REPID is applied are possibly because the total variation of mean-centered ICE curves for non-influential features are rather small, and hence, relative loss reduction values might be high, although the absolute values are small. A potential solution to prevent these outliers is to extend the stop criterion by, e.g., a minimum absolute loss reduction constraint.

In the left plot in Figure 8, we analyzed the influence of the improvement parameter γ on the interaction strength. The difference between the threshold $\gamma = 0.15$, which we chose in Section 4.2, and $\gamma = 0.1$ is rather small, while it becomes more difficult to detect the smaller interactions with $\gamma = 0.2$. The smaller we choose γ to be, the deeper we split, and the less interaction variance remains in the final terminal nodes. Therefore, the obtained interaction strengths are more precise, and hence, our results seem to be more robust for different repetitions⁸. However, the deeper we split, the more final regions we obtain, which makes it more difficult to visually analyze the influence of the interactions on the marginal effect of the feature of interest. Hence, how to set the improvement parameter γ depends on the question the user would like to answer.

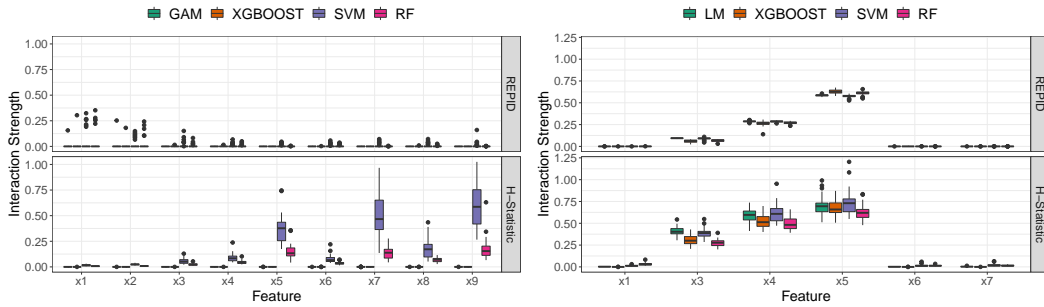


Figure 7: Left (right): The figure compares the interaction strength between x_{10} (x_2) and all other features measured by REPID and the H-Statistic for 4 different models on 30 repetitions of the described nonlinear (linear) simulation setting.

Linear example We now look at a further simulation example with only linear interaction effects between numeric features, which makes it possible to clearly rank the interactions between the feature of interest and all other features. Therefore, we draw 2000 samples of seven independent random variables, which are distributed as follows: $X_1, \dots, X_5 \sim \mathcal{U}(-1, 1)$, $X_6 \sim \mathcal{N}(0, 4)$ and $X_7 \sim \mathcal{N}(2, 9)$. The true underlying relationship is defined by $f(\mathbf{x}) = r(\mathbf{x}) + \epsilon$, where the remainder $r(\mathbf{x})$ is given by

$$r(\mathbf{x}) = x_1 + 4x_2 + 3x_2x_3 + 5x_2x_4 + 7x_2x_5$$

and $\epsilon \sim \mathcal{N}(0, (\sigma(r(\mathbf{x})) \cdot 0.1)^2)$. Hence, x_5 interacts most with x_2 , followed by x_4 and then x_3 . We fitted a linear model (LM) and an XGBOOST model with interaction constraints as well as an SVM and RF using the same configurations as for the nonlinear example on the simulated data. We repeated the experiment 30 times to quantify the interaction strength between x_2 and all other features using REPID and the H-Statistic.⁹ We use the same specifications for the models' and interaction detection methods' hyperparameters as used in Section 4.2. The right plot in Figure 7 illustrates that both methods on average find the correct ranking of the feature interactions. However, REPID shows almost no variation over all repetitions and hence leads to more stable and clearer ranking results than the H-Statistic. In the right plot of Figure 8, the impact of the improvement parameter γ is shown. However, for this example, we barely see a difference between the different choices of γ , which might be due to the simplicity of the setting and hence that no deep trees are necessary to receive stable results for the interaction strength.

⁸The more robust results are shown by smaller interquartile ranges of boxplots in Figure 8.

⁹The mean (standard deviation) of the models' test performance (measured by the mean squared error) is for the LM: 0.15 (0.002), XGBOOST: 0.6 (0.22), SVM: 0.31 (0.069) and RF: 1.43 (0.34).

REPID: Regional Effect Plots with implicit Interaction Detection

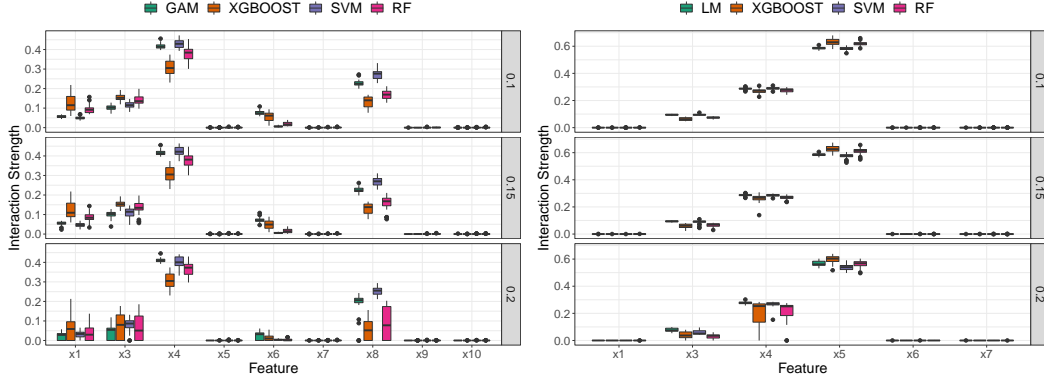


Figure 8: Left (right): The figure compares the interaction strength between x_2 and all other features measured by REPID for 3 different improvement parameter values: $\gamma = 0.1$ (top), $\gamma = 0.15$ (middle), and $\gamma = 0.2$ (bottom) for 4 different models on 30 repetitions of the described nonlinear (linear) simulation setting.

B.3 Robustness analysis

An oft-stated limitation of the usage of decision trees is that they do not provide robust results. In Section 4 and B.2, we already showed that REPID provides robust results with regards to quantifying the interaction strength for different simulation settings. To investigate the robustness itself of the fitted trees, we extract and analyze the splits of the first three levels (depths) of the tree for the nonlinear example of Section 4.2, which is the most complex analyzed example of all examples in this paper. The frequencies of the features used at each split for the 30 repetitions is shown in Table 4 for each of the fitted models. For all repetitions and for all models, x_4 was always chosen as the first splitting feature, with an average split value very close to 0, which shows only small variations (sd values). Furthermore, all models chose most often x_8 for all nodes in the second level and x_3 for all nodes in the third level of the tree. For the GAM that was correctly specified according to the true underlying function, the splits for the first three levels of the fitted decision tree barely differ. On the other hand, the SVM and the RF show higher variations. However, these models might have learned different interaction effects for different repetitions, and hence, it might be reasonable to receive different splits and REPs. The XGBOOST model also varies more than the GAM, which might be due to the fact that the GAM has a better and less variable model performance compared to the XGBOOST model, and hence, effect sizes might also vary less (see Figure 5). However, for all models, the feature chosen most often in each node is the same. It follows that REPID seems to provide robust results with regards to the interaction strength and the upper levels of the fitted tree if the same interactions have been learned by the ML models we want to explain.

B.4 Real-World Examples

Titanic dataset In Section 5, we applied REPID on the *titanic* dataset (Dawson, 1995). The labeled part of the dataset consists of 11 features and the binary survival target variable of 891 passengers. The features of the raw dataset include: *PassengerId*, *Name*, *Pclass*, *Sex*, *Age*, *SibSp*, *Parch*, *Ticket*, *Fare*, *Cabin*, *Embarked*, a detailed definition of each feature can be found at <https://www.kaggle.com/c/titanic/data>. To fit the RF model and analyze the predictions, we first pre-processed the data according to the following kaggle notebook <https://www.kaggle.com/nitinar1/titanic-solution-using-random-forest-tool-r>. The pre-processing steps can be summarized as follows:

- 1 We extract a title from the feature *Name* and categorize them into 5 categories (Master, Miss, Mr, Mrs and Rare Title).
- 2 We create a family size feature *FsizeD* from the features *Sibsp* as the number of siblings and *Parch* as the number of parents and children, and we categorize it into singleton, small and large family size.
- 3 We impute missing values of feature *Embarked* based on the fare price they paid.

Julia Herbing, Bernd Bischl, Giuseppe Casalicchio

- 4 We impute missing values of feature *Fare* by its median value of the respective *Pclass* and *Embarked* categories.
- 5 We impute the feature *Age* using a random forest imputation via multivariate imputation by chained equations.
- 6 We exclude the features *PassengerId*, *Name*, *Ticket*, *Cabin* from the dataset, which leaves us with nine features: *Pclass*, *Sex*, *Age*, *SibSp*, *Parch*, *Fare*, *Embarked*, *Title*, *FsizeD*.

California housing dataset As a second example, we applied REPID on the *California housing* dataset (Pace and Barry, 1997). The dataset contains information of a block group (small geographical unit), with an average population of around 1425 on the median house value (target), eight numeric features, and one categorical feature describing the ocean proximity. The features of the dataset include: *Longitude*, *Latitude*, *Housing median age*, *Total rooms*, *Total Bedrooms*, *Population*, *Households*, *Median Income* and *Ocean proximity*. A detailed definition of each feature can be found at <https://www.kaggle.com/camnugent/california-housing-prices>. Only the feature *Total bedroom* contains 207 missing values, which we imputed by the median value of *Total bedroom* of all other observations. Before applying the neural network on the data, we log transformed the target variable with a base of 10 and log transformed the features *Total rooms*, *Total Bedrooms*, *Population*, *Households*, *Median Income* using the natural logarithm. After pre-processing the data, we fit a neural net with one hidden layer of size 20, a weight decay of 0.1, and a maximum number of iterations of 1000. Thus, we obtain a mean absolute error (R-squared) of 0.08 (0.78) under 5-fold cross-validation. The left plot in Figure 9 shows that the median house value on average decreases the farther west a house is. The effect of individual observations seems to vary. However, visualizing ICE curves for such a high number of observations is not very insightful. In the right plot, we illustrate the resulting REPs after applying REPID with the same configurations as used for the titanic example in Section 5 but with $\gamma = 0.25$. The REPs show that the marginal effect of Longitude on the predicted median house value highly depends on how far north a house is (Latitude: the higher the value the farther north) and how close the house is to the ocean (Ocean proximity). For example, median values of houses that are farther north decrease with Longitude (light orange), while median values of houses farther south and not in the inland increase with Longitude (red).

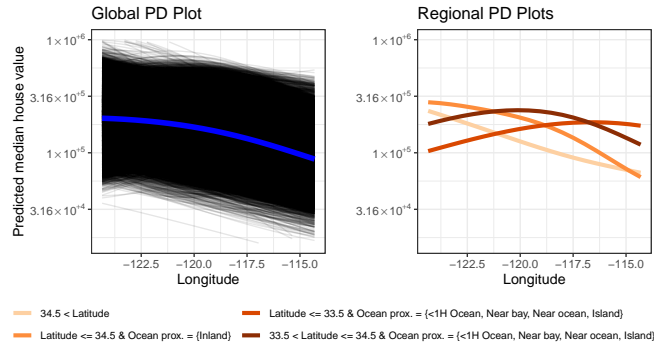


Figure 9: The figure shows the global PD plot (blue), including ICE curves (left) and the REPs after applying REPID (right) for the feature of interest Longitude of the *California housing* dataset. The interaction importance $intImp_j$ between Longitude and the interacting features is 0.49 (Latitude), 0.18 (Ocean proximity), and $R_{int}^2 = 0.67$.

Diabetes dataset As a third real-world example, we apply REPID on the *Diabetes dataset*, which analyzes diabetes in Pima Indian women and is available in the *MASS* package in R. The dataset consists of seven numeric features and the binary target variable *type*, which indicates if a woman is diabetic. The features for the 332 women contained in the dataset include: *Npreg* (number of pregnancies), *Glu* (plasma glucose concentration), *Bp* (diastolic blood pressure in mm Hg), *Skin* (triceps skin fold thickness in mm), *Bmi* (body mass index, *ped* (diabetes pedigree function), *Age*. We trained an SVM using epsilon support vector regression with a Gauss

REPID: Regional Effect Plots with implicit Interaction Detection

kernel, $C = 1$ and $\epsilon = 0.1$. Subsequently, we obtained a balanced accuracy of 0.72 using a 5-fold cross-validation. We are interested in how the feature *Skin* influences the predicted probability for diabetes. When looking at the global PDP in Figure 10, one would assume that the skin fold thickness does not effect the predicted probability for diabetes, however, the ICE curves in the left plot indicate heterogeneous effects and, hence, interactions. We apply REPID with the same configurations as used in the *titanic* example in Section 5 and obtain the REPs shown in the right plot of Figure 10. While the risk of diabetes is in general higher for women with a glucose concentration higher than 133 than for women with a lower glucose concentration, the REPs also show that the risk for women with high glucose concentration values first increases with skin fold thickness and then decreases (green and light green curves), while the risk of diabetes for women with lower glucose concentration values and a maximum of five pregnancies first slightly decreases until a thickness of approximately 20 mm and then increases with skin fold thickness (orange and red curve).

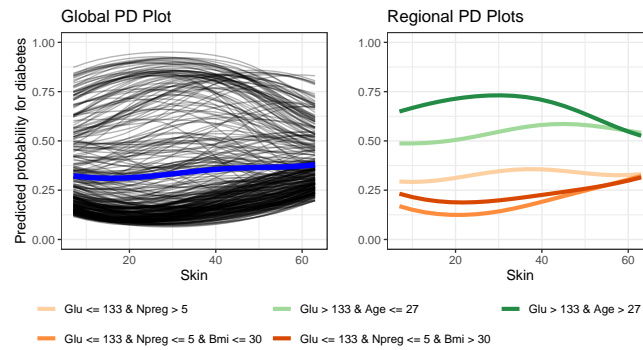


Figure 10: The figure shows the global PD plot (blue), including ICE curves (left), and the REPs after applying REPID (right) for the feature of interest *Skin* of the *diabetes* dataset. The interaction importance $intImp_j$ between *Skin* and the interacting features is 0.29 (Glu), 0.09 (Age), 0.08 (Npreg), 0.03 (Bmi) and $R_{int}^2 = 0.49$.

Julia Herbinger, Bernd Bischl, Giuseppe Casalicchio

Table 4: Summary of the split information of the first three levels (depths) of the trees fitted by applying REPID to the simulation example stated in Section 4.2 for the 30 repetitions of the 4 models (GAM, XGBOOST, SVM, RF). The column “Depth” indicates the tree depth while the column “Node ID” indicates the respective node of this depth from left to right. The columns “Feature” and “Share” provide information of how often which feature was chosen for splitting in the respective node. The last two columns contain the mean and standard deviation of the respective split value. The coloring indicates the feature that was chosen most often for each node, where the different colors belong to the different tree depths.

Model	Depth	Node ID	Feature	Share	Split value mean	Split value sd
GAM	1	1	x4	1.00	0.01	0.02
	2	1	x8	1.00	0.02	0.10
	2	2	x8	1.00	-0.00	0.10
	3	1	x3	1.00	-0.04	0.14
	3	2	x8	0.03	0.31	
	3	2	x3	0.97	0.03	0.16
	3	3	x8	0.03	-0.37	
	3	3	x3	0.97	-0.05	0.16
	3	4	x8	0.07	0.38	0.07
3	4	x3	0.93	-0.01	0.14	
XGBOOST	1	1	x4	1.00	0.00	0.01
	2	1	x8	0.63	-0.06	0.11
	2	1	x3	0.37	-0.02	0.25
	2	2	x8	0.77	-0.05	0.11
	2	2	x3	0.23	0.01	0.11
	3	1	x8	0.17	-0.06	0.16
	3	1	x3	0.63	0.03	0.14
	3	1	x1	0.20	-0.01	0.10
	3	2	x8	0.07	-0.19	0.01
	3	2	x3	0.63	0.03	0.21
	3	2	x1	0.30	0.01	0.06
	3	3	x8	0.10	0.13	0.22
	3	3	x3	0.77	-0.04	0.19
	3	3	x1	0.13	-0.02	0.06
	3	4	x8	0.03	0.00	
3	4	x3	0.77	0.08	0.20	
3	4	x1	0.20	0.06	0.07	
SVM	1	1	x4	1.00	-0.03	0.07
	2	1	x8	1.00	-0.02	0.10
	2	2	x8	1.00	-0.11	0.10
	3	1	x4	0.23	-0.45	0.06
	3	1	x8	0.03	-0.55	
	3	1	x3	0.73	-0.02	0.15
	3	2	x4	0.37	-0.50	0.09
	3	2	x3	0.63	-0.15	0.13
	3	3	x4	0.20	0.35	0.07
	3	3	x8	0.07	-0.61	0.00
3	3	x3	0.73	-0.18	0.16	
3	4	x4	0.07	0.24	0.05	
3	4	x3	0.93	-0.21	0.15	
RF	1	1	x4	1.00	0.00	0.02
	2	1	x8	0.70	-0.12	0.09
	2	1	x3	0.30	0.21	0.18
	2	2	x8	1.00	-0.11	0.15
	3	1	x8	0.23	-0.08	0.13
	3	1	x3	0.70	0.17	0.18
	3	1	x1	0.07	0.04	0.17
	3	2	x8	0.30	-0.18	0.17
	3	2	x3	0.70	0.20	0.18
	3	3	x8	0.03	-0.48	
	3	3	x3	0.97	0.10	0.18
3	4	x3	0.97	0.08	0.23	

6. Decomposing Global Feature Effects Based on Feature Interactions

This article also deals with the aggregation bias of global feature effect methods due to feature interactions. Here, we introduce the general framework *generalized additive decomposition of global effects* (GADGET) based on recursive partitioning to minimize the feature interactions between any set of features and thus to additively decompose their joint effect into the features' main effects within the found regions. Compared to the REPID method suggested in the contributing article of Section 5, GADGET is applicable to many global feature effect methods, including PD, ALE, and SHAP dependence and to multiple features of interest. We also show that the REPID method is a special case of GADGET.

Contributing article: Herbinger, J., Bischl, B., and Casalicchio, G. (2023). Decomposing global feature effects based on feature interactions. *arXiv preprint arXiv:2306.00541*. Under Review at the Journal of Machine Learning Reserach (JMLR).

Author contributions: Julia Herbinger contributed to this paper as the first author with the following contributions:

Julia Herbinger developed the project idea and the algorithms. Julia Herbinger provided the mathematical foundation and proofs in the paper, which were revised by Giuseppe Casalicchio. Julia Herbinger designed and conducted the experiments as well as the real-world applications. Julia Herbinger created all visualizations and drafted the entire manuscript. All authors contributed to revisions of the paper. Giuseppe Casalicchio and Bernd Bischl gave valuable input throughout the project and suggested several notable modifications.

Decomposing Global Feature Effects Based on Feature Interactions

Julia Herbinger

JULIA.HERBINGER@STAT.UNI-MUENCHEN.DE

Bernd Bischl

BERND.BISCHL@STAT.UNI-MUENCHEN.DE

Giuseppe Casalicchio

GIUSEPPE.CASALICCHIO@STAT.UNI-MUENCHEN.DE

Department of Statistics, LMU Munich

Munich Center for Machine Learning (MCML)

Editor:

Abstract

Global feature effect methods, such as partial dependence plots, provide an intelligible visualization of the expected marginal feature effect. However, such global feature effect methods can be misleading, as they do not represent local feature effects of single observations well when feature interactions are present. We formally introduce *generalized additive decomposition of global effects* (GADGET), which is a new framework based on recursive partitioning to find interpretable regions in the feature space such that the interaction-related heterogeneity of local feature effects is minimized. We provide a mathematical foundation of the framework and show that it is applicable to the most popular methods to visualize marginal feature effects, namely partial dependence, accumulated local effects, and Shapley additive explanations (SHAP) dependence. Furthermore, we introduce a new permutation-based interaction test to detect significant feature interactions that is applicable to any feature effect method that fits into our proposed framework. We empirically evaluate the theoretical characteristics of the proposed methods based on various feature effect methods in different experimental settings. Moreover, we apply our introduced methodology to two real-world examples to showcase their usefulness.

Keywords: interpretable machine learning, feature interactions, partial dependence, accumulated local effect, SHAP dependence

1 Introduction

Machine learning (ML) models are increasingly used in various application fields—such as medicine (Shipp et al., 2002) or social sciences (Stachl et al., 2020)—due to their better predictive performance compared to simpler, inherently interpretable models. The superior performance often comes from complex non-linear relationships or feature interactions in the data which can be modeled more accurately by more flexible and complex ML models. However, the more complex and flexible a model, the harder it becomes to explain its inner workings. A lack of explainability might hurt trust or might even be a deal-breaker for high-stakes decisions (Lipton, 2018). Hence, ongoing research on model-agnostic interpretation methods to explain any ML model has grown quickly in recent years.

One promising type of explanation is produced by feature effect methods which explain how features influence the model predictions (similarly to the coefficients in a linear model) (Molnar, 2022). We distinguish between local and global feature effect methods. Local

feature effect methods—such as individual conditional expectation (ICE) curves (Goldstein et al., 2015) or Shapley values / Shapley additive explanations (SHAP) (Štrumbelj and Kononenko, 2014; Lundberg and Lee, 2017)—explain how each feature influences the prediction of a single observation. In contrast, global feature effect methods explain the general model behavior based on the given data. Since global feature effects of ML models are often non-linear, it is easier to visualize them as done by partial dependence (PD) plots (Friedman, 2001), accumulated local effects (ALE) plots (Apley and Zhu, 2020), or SHAP dependence (SD) plots (Lundberg et al., 2019).

Aggregating individual explanations to a global explanation (e.g., ICE to PD curves) has the advantage that the global feature effects can be presented in an easy-to-understand way. However, the aggregation step might cause information loss due to heterogeneity in local effects (e.g., see the different shapes of ICE curves in Figure 1). This so-called aggregation bias is usually induced by feature interactions leading to a global feature effect that is not representative for many individuals in the data (Herbinger et al., 2022; Mehrabi et al., 2021). We term this heterogeneity *interaction-related heterogeneity*. Therefore, global explanations might be misleading or not give a complete picture when feature interactions are present, as illustrated in the bikesharing example in Figure 1 (see also Section 7). This

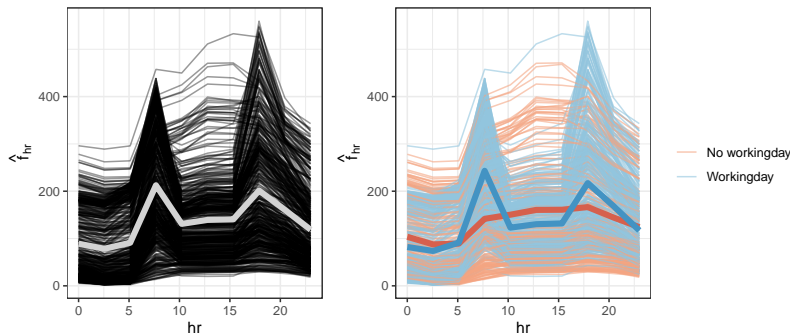


Figure 1: Left: ICE and global PD curves of feature hr (hour of the day) of the bikesharing data set (James et al., 2022). Right: ICE and regional PD curves of hr depending on feature $workingday$. The feature effect of hr on predicted bike rentals is different on working days compared to non-working days, which is due to aggregation not visible in the global feature effect plot (left).

is particularly relevant when ML models are trained on biased data (Mehrabi et al., 2021). The ML model might learn this bias, which might not be visible in global explanations due to aggregation (e.g., see COMPAS example in Section 7).

To bridge the gap between local and global effect explanations, so-called subgroup approaches that partition the feature space to obtain meaningful regional explanations within each partition have recently been introduced (e.g., Hu et al., 2020; Molnar et al., 2023; Scholbeck et al., 2022; Herbinger et al., 2022). Herbinger et al. (2022) introduced a recursive partitioning algorithm that finds interpretable subgroups in the data where the feature

DECOMPOSING GLOBAL FEATURE EFFECTS

interactions between a specific feature of interest and other features are minimized based on ICE curves. Thus, the resulting regional PD plots of the feature of interest are more representative for the individuals within the respective subgroup. However, the method is limited to PD plots with one feature of interest and, hence, leads to different partitions if multiple features of interest are considered. The mathematical foundation of their method relies on the functional ANOVA decomposition (Stone, 1994; Hooker, 2007) of the prediction function. Being able to decompose the predictions into main and higher-order effects is very appealing to better understand how features individually and jointly influence the predictions. However, the decomposition might not be unique and the respective estimation complex in the presence of feature interactions (Hooker, 2007; Lengerich et al., 2020).

Contributions. We introduce the framework GADGET, which partitions the feature space into interpretable subspaces by minimizing feature interactions based on some feature effect method. We prove that the objective of GADGET minimizes feature interactions of any feature subset and for any feature effect method that satisfies the *local decomposability* axiom (Section 4.1 and 4.2). We show that the most popular feature effect methods (PD, ALE, and SD) satisfy the *local decomposability* axiom. For each method, we introduce an estimation and visualization technique for the regional feature effect and the interaction-related heterogeneity (Section 4.3-4.5). Moreover, we propose several measures to quantify feature interactions based on GADGET, which provide more insights into the learned effects and remaining interaction-related heterogeneity (Section 4.6). To the best of our knowledge, we are the first who introduce such a flexible framework for more insights into regional feature effects and feature interactions. Additionally, we introduce the permutation interaction test (PINT) algorithm to detect significant feature interactions based on the underlying feature effect method (Section 5). Finally, we empirically evaluate the theoretical characteristics of the different methods based on several simulation settings (Section 6).

Open Science and Reproducibility. The implementation of the proposed methods as well as reproducible scripts for the experiments are provided in Online Appendix 1.

2 Background and Related Work

In this section, we introduce relevant notation and summarize related work as well as the required methodological background for this paper.

2.1 General Notation

We consider a feature space $\mathcal{X} \in \mathbb{R}^p$ and a target space \mathcal{Y} that for instance, in the case of regression is $\mathcal{Y} = \mathbb{R}$. The random variables for the features are denoted by $X = (X_1, \dots, X_p)$ and Y for the target variable. The realizations of these random variables are sampled i.i.d. from the joint probability distribution $\mathbb{P}_{X,Y}$ (which is unknown) and are denoted by $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$. The i -th observation of \mathcal{D} is denoted by $\mathbf{x}^{(i)} = (x_1^{(i)}, \dots, x_p^{(i)})^T$ and the j -th feature by $\mathbf{x}_j = (x_j^{(1)}, \dots, x_j^{(n)})^T$. A true function f maps the feature space to the target space $f : \mathcal{X} \rightarrow \mathcal{Y}$. In ML, we strive to approximate this true relationship by a

prediction model \hat{f} , which is learned on \mathcal{D} . Furthermore, we denote $-j = \{1, \dots, p\} \setminus j$ to be the set of all features besides feature j .

2.2 Functional ANOVA Decomposition

The functional ANOVA decomposition has already been studied by Stone (1994) to explain prediction models. Hooker (2004, 2007), Rahman (2014), and Li and Rabitz (2012) (amongst others) generalized the approach, suggested further estimation algorithms, and provided several analyses with regard to decomposing an ML prediction function into so-called main and higher-order (interaction) effects. If the influence of the feature \mathbf{x}_j on the prediction function cannot solely be described by the main effect of \mathbf{x}_j because the prediction changes depending on another feature \mathbf{x}_k , then the prediction function exhibits an interaction between the features \mathbf{x}_j and \mathbf{x}_k . More formally, Friedman and Popescu (2008) defined the presence of an interaction between two features \mathbf{x}_j and \mathbf{x}_k by $\mathbb{E} \left[\frac{\delta^2 \hat{f}(X)}{\delta X_j \delta X_k} \right]^2 > 0$. If the feature \mathbf{x}_j does not interact with any other feature in \mathbf{x}_{-j} , the prediction function $\hat{f}(\mathbf{x})$ can be additively decomposed into a function $h_j(\mathbf{x}_j)$ that only depends on feature \mathbf{x}_j and another function $h_{-j}(\mathbf{x}_{-j})$ that only depends on all other features \mathbf{x}_{-j} , i.e., $\hat{f}(\mathbf{x}) = h_j(\mathbf{x}_j) + h_{-j}(\mathbf{x}_{-j})$.

The prediction function $\hat{f}(\mathbf{x})$ (if it is square-integrable) can be decomposed by using the functional ANOVA decomposition as follows:

$$\hat{f}(\mathbf{x}) = g_0 + \sum_{j=1}^p g_j(\mathbf{x}_j) + \sum_{j \neq k} g_{jk}(\mathbf{x}_j, \mathbf{x}_k) + \dots + g_{12\dots p}(\mathbf{x}) = \sum_{k=1}^p \sum_{\substack{W \subseteq \{1, \dots, p\}, \\ |W|=k}} g_W(\mathbf{x}_W), \quad (1)$$

where g_0 represents a constant, $g_j(\mathbf{x}_j)$ denotes the main effect of the j -th feature, $g_{jk}(\mathbf{x}_j, \mathbf{x}_k)$ is the pure two-way interaction effect between features \mathbf{x}_j and \mathbf{x}_k , and so forth. The last component $g_{12\dots p}(\mathbf{x})$ contains the residual term, which always allows for an exact decomposition.

We will now introduce the standard functional ANOVA decomposition and a generalization of it that is more suitable in the context of correlated features.

Standard Functional ANOVA Decomposition. The standard functional ANOVA decomposition (Hooker, 2004; Rahman, 2014) assumes that all features are independent. Hence, the joint probability density function $w(\mathbf{x})$ can be written as a product-type probability measure $w(\mathbf{x}) = \prod_{j=1}^p w_j(\mathbf{x}_j)$, with $w_j : \mathbb{R} \rightarrow \mathbb{R}_0^+$ being the marginal probability density function of the j -th feature. In this case, the component functions $g_W(\mathbf{x}_W)$ can be optimally and uniquely defined for a fixed prediction function $\hat{f}(\mathbf{x})$ if the *vanishing condition* is satisfied. The *vanishing condition* is given by $\int g_W(\mathbf{x}_W) w_j(\mathbf{x}_j) d\mathbf{x}_j = 0 \quad \forall j \in W \neq \emptyset$, with $w_j(\mathbf{x}_j) \geq 0$ and $\int_{\mathbb{R}} w_j(\mathbf{x}_j) d\mathbf{x}_j = 1$ (see, e.g., Li and Rabitz, 2012; Rahman, 2014, for a detailed definition¹). Following from that, the component functions can be determined sequentially by

$$g_W(\mathbf{x}_W) = \int_{\mathbf{x}_{-W}} \left(\hat{f}(\mathbf{x}) w(\mathbf{x}) - \sum_{V \subset W} g_V(\mathbf{x}_V) \right) d\mathbf{x}_{-W}. \quad (2)$$

1. In Rahman (2014) this is called the strong annihilating condition.

DECOMPOSING GLOBAL FEATURE EFFECTS

Generalized Functional ANOVA Decomposition. When features are not independent from each other, the *vanishing condition* does not hold. Therefore, Hooker (2007) introduces a relaxed version of the *vanishing condition*² that leads to a unique decomposition of $\hat{f}(\mathbf{x})$, which they call the weighted functional ANOVA. The functional components cannot be determined sequentially (as in Eq. 2) and must be determined by solving a computationally more expensive optimization problem.

Thus, it is possible to decompose the prediction function in many different (valid) ways, with differing interpretations in the presence of feature interactions. Functional ANOVA decomposition is one possibility, which always leads to a unique decomposition. Hence, recent research has focused either on proposing models that directly include the respective conditions in the optimization process by constraints (e.g., Sun et al., 2022) or on purifying the resulting decomposition for specific models within the modelling process (e.g., Lengerich et al., 2020; Hu et al., 2022). While these approaches are first steps for computing the generalized functional ANOVA decomposition more efficiently, estimating the true underlying data distribution remains an open challenge that also influences the decomposition itself (Lengerich et al., 2020). First attempts in this direction have been proposed, for example by Sun et al. (2022) using an adaptive kernel method. Note that the estimation of the generalized functional ANOVA decomposition is only complex in the presence of feature interactions, otherwise the prediction function can easily and uniquely be decomposed into the main effects of each feature—e.g., estimated by a generalized additive model (GAM).

2.3 Visualizing Feature Effects

The visualization of feature effects provides a better understanding of how features individually or jointly influence the predicted outcome of an ML model. In this section, we introduce some of the most important global feature effect methods and relate them to the functional ANOVA decomposition.

Partial Dependence. The *PD plot* introduced by Friedman (2001) visualizes the marginal effect of a set of features $W \subset \{1, \dots, p\}$ by integrating over the joint distribution over all other features in $-W = \{1, \dots, p\} \setminus W$, which we denote by $\mathbb{P}(\mathbf{x}_{-W})$. Therefore, the PD function is defined by

$$f_W^{PD}(\mathbf{x}_W) = E_{X_{-W}}[\hat{f}(\mathbf{x}_W, X_{-W})] = \int \hat{f}(\mathbf{x}_W, \mathbf{x}_{-W}) d\mathbb{P}(\mathbf{x}_{-W}). \quad (3)$$

As the joint distribution $\mathbb{P}(\mathbf{x}_{-W})$ is usually unknown, the PD function is estimated using Monte-Carlo integration by $\hat{f}_W^{PD}(\mathbf{x}_W) = \frac{1}{n} \sum_{i=1}^n \hat{f}(\mathbf{x}_W, \mathbf{x}_{-W}^{(i)})$. Since the PD function is usually estimated for visualization purposes, the number of features in W is chosen to be one or two, and is visualized by the pairs $\{(\mathbf{x}_W^{(k)}, \hat{f}_W^{PD}(\mathbf{x}_W^{(k)}))\}_{k=1}^m$ for m grid points³.

The PD curve for $|W| = 1$ averages over heterogeneous effects that are induced by feature interactions between the feature \mathbf{x}_W and other features. These heterogeneous effects can be visualized using *ICE plots* (Goldstein et al., 2015), which measure the extent to which the

2. The relaxed vanishing condition is given by $\int_{\mathbb{R}} g_W(\mathbf{x}_W) w_W(\mathbf{x}_W) d\mathbf{x}_j = 0$ for $j \in W \neq \emptyset$, with $w(\mathbf{x})$ being a general probability density.

3. Instead of using all feature values \mathbf{x}_W , an equidistant grid or a grid based on randomly selected feature values or quantiles of \mathbf{x}_W are commonly chosen (Molnar et al., 2022).

prediction of each observation changes when the value of feature \mathbf{x}_W changes. Thus, ICE plots visualize each individual curve $\{(\mathbf{x}_W^{(k)}, \hat{f}(\mathbf{x}_W^{(k)}, \mathbf{x}_{-W}^{(i)}))\}_{k=1}^m$ for all $i \in \{1, \dots, n\}$, with the PD curve being the average over all ICE curves (see Figure 2).

While ICE plots can help to spot interactions between the feature of interest and other features by visual inspection, they do not reveal with which other features \mathbf{x}_W interacts and in which way these interactions influence the marginal effect of the feature of interest. To that end, Inglis et al. (2022) suggest different visualization techniques—such as Zenplots—that only show relevant 2-dimensional PD plots in a user-friendly layout. Other works focus on grouping ICE curves with similar shapes to find regions within the feature space where the regional PD plot can be interpreted reliably. For example, Britton (2019) suggests using an unsupervised clustering approach to group ICE curves based on their partial derivatives. A similar approach has been introduced by Zhang et al. (2021). However, Herbing et al. (2022) showed that this approach can produce misleading interpretations and suggest a supervised approach to find interpretable regions where feature interactions are minimized and resulting regional PD estimates are more representative for the underlying observations.

Friedman (2001) argues that using PD functions as additive components in a functional decomposition can recover the prediction function up to a constant. Thus, the prediction function can be decomposed into an intercept g_0 and the sum of mean-centered PD functions with the same sequential procedure, as done for the standard functional ANOVA decomposition where lower-order effects are subtracted:

$$\hat{f}(\mathbf{x}) = g_0 + \sum_{k=1}^p \sum_{\substack{W \subseteq \{1, \dots, p\}, \\ |W|=k}} \left(\hat{f}_W^{PD,c}(\mathbf{x}_W) - \sum_{V \subset W} \hat{f}_V^{PD,c}(\mathbf{x}_V) \right), \quad (4)$$

with $\hat{f}_W^{PD,c}(\mathbf{x}_W) = \hat{f}_W^{PD}(\mathbf{x}_W) - \frac{1}{m} \sum_{k=1}^m \hat{f}_W^{PD}(\mathbf{x}_W^{(k)})$. Tan et al. (2018) furthermore note that if the prediction function can be written as a sum of main effects, it can be exactly decomposed by an intercept plus the sum of all mean-centered 1-dimensional PD functions.

Similar to standard functional ANOVA, Hooker (2007) illustrate that the decomposition via PD functions is misleading when features are highly correlated due to placing too much weight in sparse regions, which causes extrapolation.

Extrapolation Problem and ALE. Since the marginal distribution is used in PD functions, we *extrapolate* in empty or sparse regions of the feature space that might even be unrealistic (e.g., predicting a disease status for a pregnant man) when features are highly correlated. This can lead to inaccurate PD estimates, especially in the case of non-parametric models such as neural networks (Apley and Zhu, 2020).

Rather than integrating over marginal distributions (see Eq. 3), one possible solution to this challenge is to integrate over conditional distributions, which is known as a marginal (M) plot (Friedman, 2001). However, the M plot of feature \mathbf{x}_j not only represents the feature effect of the feature itself, but also includes the partial effects of features correlated with the feature of interest (see Appendix A for an example). Therefore, we cannot additively decompose the prediction function into individual M plot components, as done for PD in Eq. (4)⁴. As we focus on methods that allow an additive decomposition of the prediction

4. This is only possible if all features are independent of each other. The M plot then results in the PD plot, since the conditional joint distribution is then equivalent to the marginal joint distribution.

DECOMPOSING GLOBAL FEATURE EFFECTS

function and the isolated interpretation of individual feature effects, we will not cover M plots (i.e., conditional PD plots) in greater detail here.

ALE plots are based on conditional expectations and thus avoid extrapolation. However, they solely reflect the influence of the feature of interest on the predictions (Apley and Zhu, 2020). The uncentered ALE $f_W^{ALE}(x)$ for $|W| = 1$ at feature value $x \sim \mathbb{P}(\mathbf{x}_W)$ and with $z_0 = \min(\mathbf{x}_W)$ is defined by

$$f_W^{ALE}(x) = \int_{z_0}^x \mathbb{E} \left[\frac{\partial \hat{f}(X)}{\partial X_W} \middle| X_W = z_W \right] dz_W = \int_{z_0}^x \int \frac{\partial \hat{f}(z_W, \mathbf{x}_{-W})}{\partial z_W} d\mathbb{P}(\mathbf{x}_{-W} | z_W) dz_W. \quad (5)$$

ALE first calculates the local derivatives that are weighted by the conditional distribution $\mathbb{P}(\mathbf{x}_{-W} | z_W)$ and then accumulates the local effects to generate a global effect curve.

Eq. (5) is usually estimated by splitting the value range of \mathbf{x}_W in intervals and calculating the partial derivatives for all observations within each interval. The partial derivatives are estimated by the differences between the predictions of the upper ($z_{k,W}$) and lower ($z_{k-1,W}$) bounds of the k -th interval for each observation. The accumulated effect up to observation x is then calculated by summing up the average partial derivatives (weighted by the number of observations $n(k)$ within each interval) over all intervals until the interval that includes observation x , which is denoted by $k(x)$:

$$\hat{f}_W^{ALE}(x) = \sum_{k=1}^{k(x)} \frac{1}{n(k)} \sum_{i: \mathbf{x}_W^{(i)} \in]z_{k-1,W}, z_{k,W}] } \left[\hat{f}(z_{k,W}, \mathbf{x}_{-W}^{(i)}) - \hat{f}(z_{k-1,W}, \mathbf{x}_{-W}^{(i)}) \right]. \quad (6)$$

For interpretability reasons, ALE curves are usually centered by the average of the uncentered ALE curve to obtain $f_W^{ALE,c}(x) = \hat{f}_W^{ALE}(x) - \int f_W^{ALE}(\mathbf{x}_W) d\mathbb{P}(\mathbf{x}_W)$.

Another advantage of ALE is that they are computationally less expensive than PD functions. Compared to M plots, ALE also satisfies the additive recovery⁵. Note that Apley and Zhu (2020) defined the W -th order ALE function by the pure W -th order (interaction) effect, as done similarly for the functional ANOVA decomposition in Eq.(1). Furthermore, Apley and Zhu (2020) show that the ALE decomposition has an orthogonality-like property, which guarantees (similar to the generalized functional ANOVA) that the following decomposition is unique:

$$\hat{f}(\mathbf{x}) = g_0 + \sum_{\substack{W \subseteq \{1, \dots, p\}, \\ |W|=k}} \hat{f}_W^{ALE,c}(\mathbf{x}_W). \quad (7)$$

For details on estimating higher-order ALE functions, we refer to Apley and Zhu (2020).

The choice of a feature effect method will typically depend on the underlying research question. Generally, one distinguishes between understanding the model behavior or understanding the data-generating process. While PD generally answers the first question, ALE is more focused on answering the second question.

5. Meaning, the prediction function can be additively decomposed into main and higher-order effects by ALE functions (such as PD functions), as defined in Eq. (7).

SHAP Dependence. Another method that quantifies feature effects on a local level are Shapley values which have been transferred from game theory to an ML context (Shapley, 1953; Štrumbelj and Kononenko, 2014). The general idea behind this method is to distribute the payout of a single prediction of an ML model fairly among all features.

The Shapley value of a feature is then defined by the fair contribution of this feature to the prediction. Furthermore, Herren and Hahn (2022) showed that the Shapley value of feature \mathbf{x}_j at feature value $x_j = \mathbf{x}_j^{(i)}$ can be decomposed according to the functional ANOVA decomposition into main and interaction effects⁶:

$$\begin{aligned} \phi_j^{(i)}(x_j) &= \sum_{k=0}^{p-1} \frac{1}{k+1} \sum_{\substack{W \subseteq -j: \\ |W|=k}} \left(\mathbb{E}[\hat{f}(x_j, X_{-j}) | X_W = \mathbf{x}_W^{(i)}] - \sum_{V \subset \{W \cup j\}} \mathbb{E}[\hat{f}(X) | X_V = \mathbf{x}_V^{(i)}] \right) \\ &= g_j^c(x_j) + \sum_{k=1}^{p-1} \frac{1}{k+1} \sum_{W \subseteq -j: |W|=k} g_{W \cup j}^c(x_j, \mathbf{x}_W^{(i)}). \end{aligned} \quad (8)$$

While Shapley values only quantify the local effect of a feature on the prediction, Lundberg et al. (2019) propose the SHAP dependence (SD) plot as an alternative to PD plots. SD plots show the global behavior by visualizing the local Shapley values for all or a sample of observations of one feature, similarly to an ICE plot (see Figure 5). Thus, feature interactions between the feature of interest and other features also influence the shape of the resulting point cloud. Lundberg et al. (2019) suggest coloring the points according to another (potentially) interacting feature. However, if feature effects are very heterogeneous due to feature interactions, it might not be possible to identify a clear trend of the marginal feature effect (as observed similarly for ICE plots).

As with PD plots, M plots, or ALE plots, the means of estimating the conditional expectation for Shapley values is an ongoing discussion in current research (see, e.g., Sundararajan and Najmi, 2020; Chen et al., 2020). The approach taken might also influence the SD plot, since the estimation of Shapley values can generally be based on either marginal sampling (i.e., an interventional approach, similar to PD plots) or conditional sampling (i.e., an observational approach, similar to M plots). While the interventional approach also bears the problem of extrapolation, the observational approach requires the estimation of the conditional data distribution, which is still a challenging task (Aas et al., 2021). Chen et al. (2020) argue that the interventional approach is not generally wrong and can be useful if we want to derive explanations that are true to the model, while the observational approach should be used when we want to extract interpretations that are true to the data.

2.4 Quantification of Interaction Effects

Visualizing feature effects is a powerful technique to obtain a better understanding of how features influence the prediction function. However, useful visualizations are usually limited to a maximum of two features. Hence, to better comprehend which feature effects might depend on other features due to interactions, we are interested in detecting and quantifying the strength of feature interactions.

6. Similar decompositions and their estimation have been introduced in Hiabu et al. (2023) and Bordt and von Luxburg (2023).

DECOMPOSING GLOBAL FEATURE EFFECTS

Based on the definition of feature interactions in Section 2.2 and on the PD function, Friedman and Popescu (2008) introduced the H-Statistic as a global interaction measure between subsets of features. To quantify the interaction strength between a feature of interest \mathbf{x}_j and all other features \mathbf{x}_{-j} , the H-Statistic is calculated by

$$\hat{\mathcal{H}}_j^2 = \frac{\sum_{i=1}^n (\hat{f}^c(\mathbf{x}^{(i)}) - \hat{f}_j^{PD,c}(\mathbf{x}_j^{(i)}) - \hat{f}_{-j}^{PD,c}(\mathbf{x}_{-j}^{(i)}))^2}{\sum_{i=1}^n (\hat{f}^c(\mathbf{x}^{(i)}))^2}. \quad (9)$$

Hence, $\hat{\mathcal{H}}_j^2$ quantifies how much of the prediction function’s variance⁷ can be attributed to the interaction between feature \mathbf{x}_j and all other features \mathbf{x}_{-j} ⁸. Eq. (9) can be adjusted to quantify interactions of different order, such as two-way interactions (see Friedman and Popescu, 2008). Other global interaction measures to quantify two-way interactions based on PDs or SHAP interaction values have been suggested by Greenwell et al. (2018) and Herbinger et al. (2022). While these methods focus on detecting and quantifying two-way feature interactions, Hooker (2004) suggests an algorithm based on the idea of functional ANOVA decomposition to detect all important higher-order terms and visualizes the feature relationships in a network graph. However, the feature interactions are not quantified and thus cannot be ranked according to their influence on the prediction.

Next to global interaction measures, there exist multiple local interaction measures that quantify feature interactions for a single observation (e.g., Lundberg et al., 2019; Tsai et al., 2023; Blücher et al., 2022; Kumar et al., 2021).

3 Motivation: REPID and its Limitations

To obtain a better understanding of how features globally affect the predictions in the presence of feature interactions, Herbinger et al. (2022) propose the REPID method, which decomposes the global PD into regional PDs such that individual effects (ICE curves) are more homogeneous within each region. This section provides a short explanation of the REPID method and illustrates its limitations, which are addressed by our new framework introduced in Section 4.

The REPID method is based on a recursive partitioning algorithm (similar to CART by Breiman et al., 1984) that splits each parent node into two child nodes until a stop criterion is met. Compared to a common decision tree, the inputs are not the observations \mathbf{x} themselves, but the ICE curves belonging to the observations of one feature of interest \mathbf{x}_j . Herbinger et al. (2022) showed that their chosen objective minimizes interaction effects between the feature of interest and all other features. Therefore, the reduction at each split quantifies the interaction strength between the feature of interest and the feature used for splitting. Thus, the method provides more representative PD estimates in interpretable regions for a feature of interest as well as a ranking of considered two-way feature interactions.

For illustration purposes, consider the following simulation example that is often used in a slightly modified form in the context of feature interactions (see e.g., Goldstein et al.,

7. The denominator of Eq. (9) represents the prediction function’s variance, where $\hat{f}^c(\mathbf{x}^{(i)}) = \hat{f}(\mathbf{x}^{(i)}) - \frac{1}{n} \sum_{i=1}^n \hat{f}(\mathbf{x}^{(i)})$.

8. If the feature \mathbf{x}_j does not interact with any other feature in \mathbf{x}_{-j} , then the mean-centered joint effect function can be additively decomposed into: $f^c(\mathbf{x}) = f_j^{PD,c}(\mathbf{x}_j) + f_{-j}^{PD,c}(\mathbf{x}_{-j})$, leading to $\hat{\mathcal{H}}_j^2 = 0$.

2015; Herbinger et al., 2022): Let $X_1, X_2, X_3 \sim \mathcal{U}(-1, 1)$ be independently distributed and the true underlying relationship be defined by $Y = 3X_1 \mathbb{1}_{X_3 > 0} - 3X_1 \mathbb{1}_{X_3 \leq 0} + X_3 + \epsilon$, with $\epsilon \sim \mathcal{N}(0, 0.09)$. We then draw 500 observations from these random variables and fit a feed-forward neural network (NN) with a single hidden layer of size 10 and weight decay of 0.001.⁹ The R^2 measured on a separately drawn test set of size 10000 following the same distribution is 0.94. The ICE and PD curves of feature \mathbf{x}_1 for the training data are shown in the left plot of Figure 2. The ICE curves clearly show that \mathbf{x}_1 influences the prediction differently depending on another feature—in this case, \mathbf{x}_3 . If we would only consider the global PD plot, the nearly-horizontal PD curve of \mathbf{x}_1 (grey line) would indicate no influence on the model’s predictions and thus is not representative for the underlying local feature effects (ICE curves). Here, REPID can be applied to the ICE curves of \mathbf{x}_1 to search for the best split point within the feature subset in $-j$ (here: \mathbf{x}_2 and \mathbf{x}_3) and minimize the feature interactions between features in j and $-j$. In this example, the best split point found is $\mathbf{x}_3 = 0$, which is also optimal according to the data-generating process. The regional PD plots found by REPID reflect the contradicting influence of feature \mathbf{x}_1 on the predictions compared to the global PD plot. The respective split reduces the interaction-related heterogeneity almost completely (by 98%). Hence, the resulting regional marginal effects of \mathbf{x}_1 can be approximated well by the respective main effects (regional PD), which are therefore more representative for the individual observations within each region.

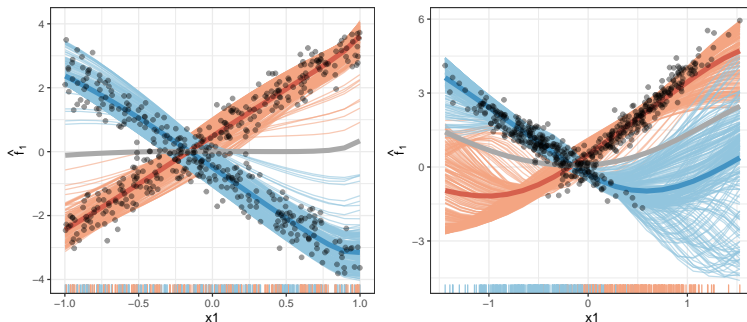


Figure 2: ICE curves and PD curves (grey) for the uncorrelated (left) and correlated case (right) of the simulation example of Section 3. ICE curves are colored w.r.t. the first split when REPID is applied. The split feature is in both cases \mathbf{x}_3 with split points -0.01 (left) and -0.15 (right). The blue color represents the left and orange the right region according to the split point. The thicker lines represent the respective regional PD curves. The rug plot shows the distribution of \mathbf{x}_1 according to the split point. The black points are the underlying observations.

9. The hyperparameters’ size (number of units of the hidden layer) and weight decay for a single-hidden-layer feed-forward NN were tuned via 5-fold cross-validation using grid search and a maximum of 1000 iterations. Considered grid values for weight decay: (0.5, 0.1, 0.01, 0.001, 0.0001, 0.00001), and for size: (3, 5, 10, 20, 30).

DECOMPOSING GLOBAL FEATURE EFFECTS

Herbinger et al. (2022) have shown that REPID provides meaningful results if 1) only one feature is of interest, and 2) if global feature effects are visualized by PD plots. These two assumptions limit the general applicability of REPID for two reasons.

First, applying REPID to different features of interest will usually result in different regions, as the method produces feature-specific regional PD plots. Hence, in the above-mentioned simulation example, REPID produces different regions when the feature of interest is \mathbf{x}_3 instead of \mathbf{x}_1 , which complicates interpretations when effects of multiple features are of interest. Moreover, we might be interested to receive regions within the feature space where interactions between multiple features are minimized, and thus the joint effect within each region can be decomposed into the main effects of the regarded features.

Second, there are other feature effect methods that might be more suitable, depending on the underlying data set and research question. Thus, a general framework that allows finding regions within the feature space where feature interactions are minimized w.r.t. an individually chosen feature effect method would be extremely useful. One main disadvantage specifically for ICE and PD plots is the extrapolation problem in the presence of feature interactions and correlations, as described in Section 2.3. To illustrate how this problem affects REPID, we consider the same simulation example as before, but with $X_1 = X_3 + \delta$ and $\delta \sim \mathcal{N}(0, 0.0625)$. We again draw 500 observations and fit an NN with the same specification and receive an R^2 of 0.92 on a separately drawn test set of size 10000 following the same distribution. Thus, the high correlation between \mathbf{x}_1 and \mathbf{x}_3 barely influences the model performance. However, Figure 2 clearly shows that ICE curves are extrapolating in low-density regions. The rug plot on the bottom indicates the distribution of feature \mathbf{x}_1 depending on feature \mathbf{x}_3 . Thus, the model was not trained on observations with small \mathbf{x}_1 values and simultaneously large \mathbf{x}_3 values, and vice versa. However, since we integrate over the marginal distribution, we also predict in these “out-of-distribution” areas. This leads to the so-called extrapolation problem and, hence, to uncertain predictions in extrapolated regions that must be interpreted with caution. In Section 6.1, we will analyze how severe the extrapolation problem is with regard to finding the correct split feature and split point.

4 Generalized Additive Decomposition of Global Effects (GADGET)

Here, we introduce a new framework called generalized additive decomposition of global effects (GADGET), which additively decomposes global feature effects based on minimizing feature interactions. Through this approach, one or multiple features (up to all p features) can be considered to find interpretable regions where feature effects of all regarded features are more representative for the underlying individual observations. We will first introduce the methodology of GADGET, which is applicable to many different feature effect methods. We formally define the axiom that must be satisfied by the desired feature effect method to be used in GADGET. We then show that the most popular feature effect methods satisfy this axiom and that REPID is a special case of GADGET.

4.1 The GADGET Algorithm

Let $h(x_j, \mathbf{x}_{-j}^{(i)})$ be the local feature effect of a feature \mathbf{x}_j of the i -th observation at some feature value $x_j \in \mathcal{X}_j$ measured by a local feature effect function $h : \mathbb{R}^p \rightarrow \mathbb{R}$, and let

$\mathbb{E}[h(x_j, X_{-j})|\mathcal{A}_g]$ be the expected feature effect of \mathbf{x}_j at x_j w.r.t. X_{-j} conditioned on the subspace $\mathcal{A}_g \subseteq \mathcal{X}$.¹⁰ Then, we can define the deviation between the local feature effect and the expected feature effect at a specific feature value x_j for subspace \mathcal{A}_g using a point-wise loss function, e.g., the squared distance¹¹:

$$\mathcal{L}(\mathcal{A}_g, x_j) = \sum_{i: \mathbf{x}^{(i)} \in \mathcal{A}_g} \left(h(x_j, \mathbf{x}_{-j}^{(i)}) - \mathbb{E}[h(x_j, X_{-j})|\mathcal{A}_g] \right)^2. \quad (10)$$

The risk function \mathcal{R} of the j -th feature and subspace \mathcal{A}_g is defined by aggregating the point-wise loss of Eq. (10) over a sample of feature values of \mathbf{x}_j :

$$\mathcal{R}(\mathcal{A}_g, \mathbf{x}_j) = \sum_{k: k \in \{1, \dots, m\} \wedge \mathbf{x}_j^{(k)} \in \mathcal{A}_g} \mathcal{L}(\mathcal{A}_g, \mathbf{x}_j^{(k)}). \quad (11)$$

For instance, feature values of size m can be sampled values of \mathbf{x}_j ¹² and their constraint depends on feasible values within the subspace \mathcal{A}_g .

The local feature effect function must be defined in such a way that the risk function in Eq. (11) minimizes the interaction-related heterogeneity of features in S , i.e., the feature interactions between $\mathbf{x}_j \in \mathbf{x}_S$ with $S \subseteq \{1, \dots, p\}$ and features of a previously defined feature set $Z \subseteq \{1, \dots, p\}$ (see Section 4.2). The feature subset S is chosen to be the subset of features for which we want to receive representative regional feature effect plots by minimizing their interaction-related heterogeneity. Features in Z are considered as split features and thus aim to partition the feature space in such a way that feature interactions with features in S are minimized. Algorithm 1 defines a single partitioning step of GADGET, which is inspired by the CART algorithm (Breiman et al., 1984) and is recursively repeated until a certain stop criterion is met (see Section 4.7). We greedily search for the best split point within the feature subset Z such that the interaction-related heterogeneity of feature subset S is minimized in the two resulting subspaces. The interaction-related heterogeneity is measured by the variance of the local feature effects of \mathbf{x}_j within the new subspace (see Eq. 11). Hence, for each split feature z and split point t , we sum up the risk of the two resulting subspaces for all features in S (line 6 in Algorithm 1) and then choose the split point of the split feature (\hat{t}, \hat{z}) that minimizes the interaction-related heterogeneity of all features $j \in S$ (line 9 in Algorithm 1).¹³

4.2 Theoretical Foundation of GADGET

To apply GADGET, a suitable local feature effect function h must be defined. General properties that GADGET requires from this function are provided by Axiom 1.

-
10. The expected value is always taken w.r.t. the random variables within the expected value. We only use a subscript for the expected value if it cannot uniquely be defined by the above notation.
 11. Other distance metrics are also possible. However, we chose the squared distance, since the interpretation in terms of variance is most intuitive in this context.
 12. The choice depends on the underlying local feature effect function h and on the data distribution. Other common choices are quantiles, or an equidistant grid from the feature range of \mathbf{x}_j .
 13. In the case where $z \in S$, we also include the heterogeneity reduction of the z -th feature in the objective, since we aim to reduce the overall interaction-related heterogeneity of all features in S .
 14. S and Z can be distinct or partially or fully overlap with each other. See Section 5 for defining S and Z .

DECOMPOSING GLOBAL FEATURE EFFECTS

Algorithm 1: Partitioning algorithm of GADGET

-
- 1: **input:** subspace $\mathcal{A} \subseteq \mathcal{X}$, risk function \mathcal{R} and feature of interest index set $S \subseteq \{1, \dots, p\}$ and feature interaction index set $Z \subseteq \{1, \dots, p\}$ ¹⁴
 - 2: **output:** subspaces $\mathcal{A}_l^{\hat{t}, \hat{z}}$ and $\mathcal{A}_r^{\hat{t}, \hat{z}}$
 - 3: **for** each feature indexed by $z \in Z$ **do**
 - 4: **for** every split t on feature \mathbf{x}_z **do**
 - 5: $\mathcal{A}_l^{t,z} = \{\mathcal{A} | \mathbf{x}_z \leq t\}$; $\mathcal{A}_r^{t,z} = \{\mathcal{A} | \mathbf{x}_z > t\}$
 - 6: $\mathcal{I}(t, z) = \sum_{j \in S} (\mathcal{R}(\mathcal{A}_l^{t,z}, \mathbf{x}_j) + \mathcal{R}(\mathcal{A}_r^{t,z}, \mathbf{x}_j))$
 - 7: **end for**
 - 8: **end for**
 - 9: Choose $(\hat{t}, \hat{z}) \in \arg \min_{t,z} \mathcal{I}(t, z)$
-

Axiom 1 (Local Decomposability) A local feature effect function $h : \mathbb{R}^p \rightarrow \mathbb{R}$ satisfies the local decomposability axiom if and only if the decomposition of the i -th local effect $h(x_j, \mathbf{x}_{-j}^{(i)})$ of feature \mathbf{x}_j at x_j solely depends on main and higher-order effects of feature \mathbf{x}_j :

$$h(x_j, \mathbf{x}_{-j}^{(i)}) = g_j(x_j) + \sum_{k=1}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=k}} g_{W \cup j}(x_j, \mathbf{x}_W^{(i)}).$$

The *local decomposability* axiom must be satisfied by the chosen local feature effect function h . Thus, the i -th local feature effect $h(x_j, \mathbf{x}_{-j}^{(i)})$ for feature \mathbf{x}_j at x_j must be defined such that it only depends on the main effect of feature \mathbf{x}_j as well as the interaction effects between \mathbf{x}_j and all other features in $-j$. This decomposition is not generally given by every local feature effect method. For example, the decomposition of ICE curves depends not only on effects including the feature of interest \mathbf{x}_j but also on effects of other features, which leads to additive shifts. However, we can usually transform the local feature effects in a meaningful manner to receive the decomposition provided in Axiom 1. ICE curves, for instance, must be mean-centered (see Appendix B.4) to satisfy the *local decomposability* axiom. If the local feature effect function satisfies Axiom 1, then Theorem 2 guarantees that the loss function defined in Eq. (10) quantifies the interaction-related heterogeneity of local effects (feature interactions) of feature \mathbf{x}_j at x_j within the regarded subspace \mathcal{A}_g .

Theorem 2 If the local feature effect function $h(x_j, \mathbf{x}_{-j}^{(i)})$ satisfies Axiom 1, then the loss function $\mathcal{L}(\mathcal{A}_g, x_j)$ defined in Eq. (10) only depends on feature interactions between the feature \mathbf{x}_j at x_j and features in $-j$:

$$\mathcal{L}(\mathcal{A}_g, x_j) = \sum_{i: \mathbf{x}^{(i)} \in \mathcal{A}_g} \left(\sum_{k=1}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=k}} g_{W \cup j}(x_j, \mathbf{x}_W^{(i)}) - \mathbb{E}[g_{W \cup j}(x_j, X_W) | \mathcal{A}_g] \right)^2.$$

The proof can be found in Appendix B.1.

Since we only use features in Z for splitting and we aggregate the resulting risk over all features $j \in S$, the objective function in GADGET minimizes the feature interactions between all features in S and interacting features in Z (see Theorem 3). Furthermore, if Z contains all features interacting with any feature in S (i.e., no feature in $-Z$ interacts with any feature in S), then the theoretical minimum of the objective minimized in Algorithm 1 is zero (see Corollary 4). This means that all feature interactions present in feature effects of the features in S can be reduced such that only main effects of these features remain in each subspace. Thus, the joint feature effect $f_{S|\mathcal{A}_g}$ within each subspace \mathcal{A}_g can be uniquely and additively decomposed into the univariate feature effects $f_{j|\mathcal{A}_g}$:

$$f_{S|\mathcal{A}_g}(\mathbf{x}_S) = \sum_{j \in S} f_{j|\mathcal{A}_g}(\mathbf{x}_j). \quad (12)$$

However, if Z is chosen such that the features contained in its complement $-Z$ interact with features contained in S , then the theoretical minimum of the objective is larger than 0. Thus, heterogeneous effects due to feature interactions between features in S and features in $-Z$ remain. The approach to choose the subsets S and Z is discussed in Section 5.

Theorem 3 *If the local feature effect function h satisfies Axiom 1 and if all features contained in Z and all features in $-Z = Z^c$ are pairwise independent, then the objective function $\mathcal{I}(t, z)$ of Algorithm 1 based on the loss function in Eq. (10) minimizes feature interactions between features within the subset S and features in Z , but does not generally minimize feature interactions between features in S and $-Z$. Since the partitions found by the GADGET algorithm to minimize the feature interactions of S only depend on features in Z and are independent of features in $-Z$, interactions between each $j \in S$ and features in $-Z$ are independent of the partitioning in Algorithm 1:*

$$\left(\sum_{l=1}^{|-Z \setminus j|} \sum_{\substack{-Z_l \subseteq -Z \setminus j, \\ |-Z_l|=l}} g_{-Z_l \cup j}(x_j, \mathbf{x}_{-Z_l}^{(i)}) - \mathbb{E}[g_{-Z_l \cup j}(x_j, X_{-Z_l}) | \mathcal{A}_b^{t,z}] \right) \perp\!\!\!\perp \mathcal{A}_b^{t,z}.$$

The proof can be found in Appendix B.2.

Corollary 4 *If the local feature effect function h satisfies Axiom 1 and if the feature subset $-Z = Z^c$ does not contain any features interacting with any $j \in S$, then the theoretical minimum of the objective function in Algorithm 1 is $\mathcal{I}(t^*, z^*) = 0$.*

The proof can be found in Appendix B.3.

The fulfillment of the *local decomposability* axiom depends on the definition of the underlying local feature effect function h . In the following sections, we show the validity of this axiom for common feature effect methods and provide estimates as well as visualizations for the resulting regional effect curves and their remaining interaction-related heterogeneity.

4.3 GADGET-PD

Here, we show the applicability of PD as feature effect method within the GADGET algorithm which we call the GADGET-PD.

DECOMPOSING GLOBAL FEATURE EFFECTS

Method. The PD plot is based on ICE curves (local feature effects) and one of the most popular global feature effect methods. However, ICE curves of feature \mathbf{x}_j do not satisfy Axiom 1, since the decomposition of the i -th ICE curve also contains main or interaction effects of the i -th observation that are independent of feature \mathbf{x}_j (see Appendix B.4). These effects can be cancelled out by centering ICE curves w.r.t. the mean of each curve (i.e., $\mathbb{E}[\hat{f}(X_j, \mathbf{x}_{-j}^{(i)})]$). The resulting mean-centered ICE curves satisfy Axiom 1 (see Appendix B.4). Hence, they are chosen as local feature effect method within GADGET: $h(x_j, \mathbf{x}_{-j}^{(i)}) = \hat{f}^c(x_j, \mathbf{x}_{-j}^{(i)}) = \hat{f}(x_j, \mathbf{x}_{-j}^{(i)}) - \mathbb{E}[\hat{f}(X_j, \mathbf{x}_{-j}^{(i)}) | \mathcal{A}_g]$.

The loss function used within GADGET-PD to minimize the interaction-related heterogeneity is then defined by the variability of mean-centered ICE curves:

$$\mathcal{L}^{PD}(\mathcal{A}_g, x_j) = \sum_{i: \mathbf{x}^{(i)} \in \mathcal{A}_g} \left(\hat{f}^c(x_j, \mathbf{x}_{-j}^{(i)}) - \mathbb{E}[\hat{f}^c(x_j, X_{-j}) | \mathcal{A}_g] \right)^2. \quad (13)$$

By choosing mean-centered ICE curves as local feature effect function h , the loss function in Eq. (13) for GADGET-PD results in the same loss function used within REPID. In Appendix B.4, we show that the REPID method is—for this specific loss function—a special case of GADGET-PD, where we have only one feature of interest (i.e., $S = j$) and where we consider all other features to be potential split features (i.e., $Z = -j$).

Note that since REPID never splits with regard to the visualized feature of interest \mathbf{x}_j , the constant used for mean-centering (i.e., $\mathbb{E}[\hat{f}(X_j, \mathbf{x}_{-j}^{(i)}) | \mathcal{A}_g] = \mathbb{E}[\hat{f}(X_j, \mathbf{x}_{-j}^{(i)})]$) always stays the same. In contrast, for the more general GADGET algorithm, the mean-centering constant depends on how S and Z are defined. Thus, the mean-centering constants of features in S might change if we also use them for splitting. For example, in Figure 3, we use feature \mathbf{x}_3 as a feature of interest in S and as a split feature in Z . In this case, the range of the visualized feature \mathbf{x}_3 is also split according to the split point found by the GADGET algorithm. Hence, the expected value conditioned on the new subspace changed (i.e., $\mathbb{E}[\hat{f}(X_j, \mathbf{x}_{-j}^{(i)}) | \mathcal{A}_g] \neq \mathbb{E}[\hat{f}(X_j, \mathbf{x}_{-j}^{(i)})]$) and thus must be adjusted to avoid additive (non-interaction) effects in the new subspace.

Illustration. Figure 3 visualizes the result when applying GADGET-PD on the uncorrelated simulation example of Section 3 by choosing $S = Z = \{1, 2, 3\}$. Hence, we are interested in the feature effect of all available features and consider all features as possible interaction (split) features. GADGET-PD performs one split with regard to $\mathbf{x}_3 \approx 0$. Thus, the correct split feature and its corresponding split point are found by GADGET such that the interaction-related heterogeneity of all features in S is almost completely reduced and only main effects within the subspaces remain.

Decomposition. If Z contains all features interacting with features in S and if GADGET is applied such that the theoretical minimum of the objective function is reached, then according to Corollary 4, the joint mean-centered PD function $f_{S|\mathcal{A}_g}^{PD,c}$ within each final subspace \mathcal{A}_g can be decomposed into the respective 1-dimensional mean-centered PD functions:

$$f_{S|\mathcal{A}_g}^{PD,c}(\mathbf{x}_S) = \sum_{j \in S} f_{j|\mathcal{A}_g}^{PD,c}(\mathbf{x}_j). \quad (14)$$

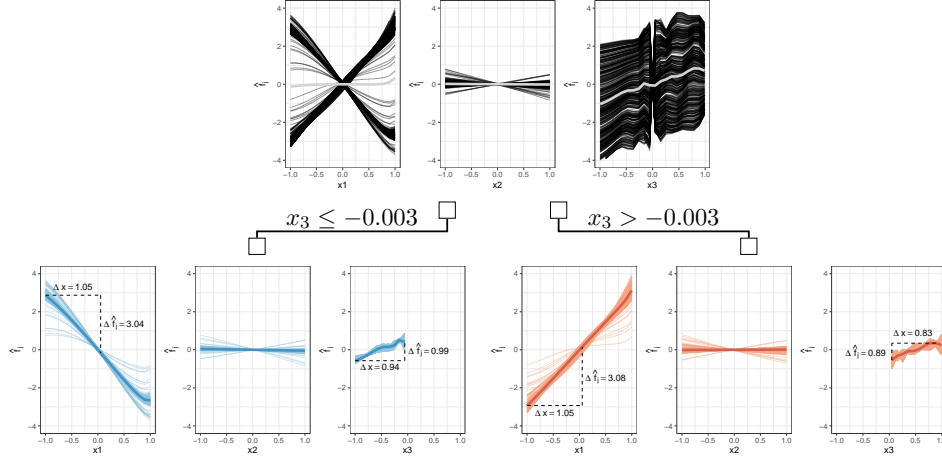


Figure 3: Visualization of applying GADGET with $S = Z = \{1, 2, 3\}$ to mean-centered ICE curves of the uncorrelated simulation example of Section 3 with $Y = 3X_1 \mathbb{1}_{X_3 > 0} - 3X_1 \mathbb{1}_{X_3 \leq 0} + X_3 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.09)$. The upper plots show the mean-centered ICE and PD curves on the entire feature space, while the lower plots represent the respective mean-centered ICE and PD curves after partitioning the feature space w.r.t. $\mathbf{x}_3 = -0.003$.

Eq. (14) is justified by the assumption that no more interactions between features in S and other features are present in the final regions (Friedman and Popescu, 2008).

Furthermore, if the subset $-S$ is the subset of features that do not interact with any other features (local feature effects are homogeneous), then according to Eq. (14) and Eq. (4), the prediction function $\hat{f}_{\mathcal{A}_g}$ within each final subspace \mathcal{A}_g can be decomposed into the 1-dimensional mean-centered PD functions of all p features plus a constant value g_0 :

$$\hat{f}_{\mathcal{A}_g}(\mathbf{x}) = g_0 + \sum_{j=1}^p f_{j|\mathcal{A}_g}^{PD,c}(\mathbf{x}_j).$$

Thus, depending on how we choose the subsets S and Z and the extent to which we are able to minimize the interaction-related heterogeneity of feature effects by recursively applying Algorithm 1, we might be able to approximate the prediction function by an additive function of main effects of all features within the final regions.

This can also be shown for the simulation example illustrated in Figure 3, where $\hat{f}_{2|\mathcal{A}_g}^{PD,c}(\mathbf{x}_2) = 0$ (the regional effect of feature \mathbf{x}_2 after the split is still 0 and has low interaction-related heterogeneity). Instead, the regional effects of \mathbf{x}_1 and \mathbf{x}_3 vary compared to the root node and strongly reduce the interaction-related heterogeneity. Since the regional effects of \mathbf{x}_1 and \mathbf{x}_3 are approximately linear, we can estimate the prediction function

DECOMPOSING GLOBAL FEATURE EFFECTS

within each subspace by

$$\hat{f}_{\mathcal{A}_l}(\mathbf{x}) = g_0 + \frac{-3.04}{1.05}\mathbf{x}_1 + \frac{0.99}{0.94}\mathbf{x}_3 = g_0 - 2.9\mathbf{x}_1 + 1.05\mathbf{x}_3$$

and

$$\hat{f}_{\mathcal{A}_r}(\mathbf{x}) = g_0 + \frac{3.08}{1.05}\mathbf{x}_1 + \frac{0.89}{0.83}\mathbf{x}_3 = g_0 + 2.93\mathbf{x}_1 + 1.07\mathbf{x}_3,$$

which is a close approximation to the underlying data-generating process and thus provides a better understanding of how the features of interest influence the prediction function compared to only considering the global PD plots.

Note that besides the decomposability property in Eq. (14), each global PD of features in S is a weighted additive combination of the final regional PDs. Thus, each global PD can be additively decomposed into regional PD.

Estimates and Visualization. To estimate and visualize the expected regional effect, one can choose between the regional PD curve $\hat{f}_{j|\mathcal{A}_g}^{PD}$ or its mean-centered version $\hat{f}_{j|\mathcal{A}_g}^{PD,c}$ for each feature $j \in S$, which are calculated by Monte-Carlo integration for each final region \mathcal{A}_g . The interaction-related heterogeneity for each feature in S and final subspace \mathcal{A}_g is measured by the risk function in Eq. (11) with the loss function in Eq. (13). Thus, the interaction-related heterogeneity quantifies the variation of mean-centered ICE curves within each region. For visualization purposes, we calculate the 95% intervals of interaction-related heterogeneity based on this variation. We then suggest a plot for each feature in S that shows the final regional PD curves and 95% interaction-related heterogeneity intervals (see e.g., Figure 11).

The main issue with local ICE curves and resulting global PD plots is the extrapolation problem when features are correlated, as demonstrated in Section 3. This problem remains for GADGET when mean-centered ICE curves are chosen as local feature effect function h .

4.4 GADGET-ALE

An alternative global feature effect method that allows an additive decomposition of the prediction function and the interpretation of individual feature effects are ALE plots (Apley and Zhu, 2020), which we summarized in Section 2.3. Here, we show their applicability within the GADGET algorithm which we term GADGET-ALE.

Method. While ALE curves compared to PD curves do not suffer from extrapolation, they do not directly entail a local feature effect visualization that shows the heterogeneity induced by feature interactions as ICE curves for PD plots. However, ALE plots are also based on corresponding local feature effects that provide information about the underlying interaction-related heterogeneity and that can be used within the GADGET algorithm to receive more representative ALE curves in the final regions. These local feature effects are the derivatives w.r.t. the feature of interest \mathbf{x}_j of the prediction function (see Eq. 5). In Appendix B.5, we show that by choosing $h = \frac{\partial \hat{f}(x_j, \mathbf{x}_{-j}^{(i)})}{\partial x_j}$, Axiom 1 is met, which leads to the following loss function used within the objective in GADGET-ALE:

$$\mathcal{L}^{ALE}(\mathcal{A}_g, x_j) = \sum_{\substack{i: \mathbf{x}^{(i)} \in \mathcal{A}_g \wedge \\ \mathbf{x}^{(i)} \in \mathbb{P}(\mathbf{x}_{-j}|x_j)}} \left(\frac{\partial \hat{f}(x_j, \mathbf{x}_{-j}^{(i)})}{\partial x_j} - \mathbb{E} \left[\frac{\partial \hat{f}(X_j, X_{-j})}{\partial x_j} \middle| \mathcal{A}_g \wedge X_j = x_j \right] \right)^2. \quad (15)$$

The derivatives are calculated by defining m intervals for feature \mathbf{x}_j and quantifying for each interval the prediction difference between the upper and lower boundary for observations lying in this interval (see Eq. 6). Hence, the loss function in Eq. (15) measures the variance of the derivatives of observations where the feature values of \mathbf{x}_j lie within the boundaries of the regarded interval and, thus, measures the interaction-related heterogeneity of local effects of feature \mathbf{x}_j within this interval. The risk function in Eq. (11) then aggregates this loss over all m intervals. Note that the conditional expectation in Eq. (15) is the expected conditional derivative (estimated by the average derivative within the regarded interval) and not the ALE curve itself. However, the ALE curve is calculated by integrating the expected conditional derivative up to the regarded value x_j (see Eq. 5).

Illustration. Figure 4 illustrates the first split for the two simulation examples of Section 3 using GADGET-ALE. The heterogeneity of the local effects (derivatives) before applying GADGET is very high, spanning across negative to positive values (grey boxplots) within each interval. With GADGET, we then partition the feature space w.r.t. one of the features in Z such that this interaction-related heterogeneity is minimized. When GADGET-ALE is used, for both the uncorrelated and the correlated case, we receive the correct split feature and approximately the correct split point, which clearly shows a high reduction in heterogeneity of these local effects after the first split. While the shapes of the centered ALE curves look very similar to PD plots (see Figure 2) for the uncorrelated case, the ALE curves for the correlated case do not extrapolate and, thus, are more representative for the feature effect with regard to the underlying data distribution.

Decomposition Equivalently to PD plots, ALE plots also contain an additive recovery and, thus, can be decomposed additively into main and interaction effects (see Section 2.3). Furthermore, if Z is defined such that all features interacting with features in S are included and if GADGET is applied so that the theoretical minimum of the objective function is reached, then—according to Corollary 4—the joint mean-centered ALE function $\hat{f}_{S|\mathcal{A}_g}^{ALE,c}$ within each final subspace \mathcal{A}_g can be decomposed into the 1-dimensional mean-centered ALE functions of features in S (see Eq. 12). Thus, for ALE plots, we might also be able to decompose the prediction function into the regional features’ main effects, depending on how we choose the subsets S and Z within the GADGET algorithm. More details on the decomposition when using GADGET-ALE and an exemplary illustration of the uncorrelated simulation example of Section 3 can be found in Appendix C.1.

Estimates and Visualization. The regional effect is estimated by the regional centered ALE curve $\hat{f}_{j|\mathcal{A}_g}^{ALE,c}$ for each feature in S and is calculated as in Eq. (6) for each final subspace \mathcal{A}_g . The interaction-related heterogeneity for each feature in S and final subspace \mathcal{A}_g is measured by the risk function in Eq. (11) with the loss function in Eq. (15). Thus, the interaction-related heterogeneity quantifies the variation of partial derivatives within each subspace. Since the partial derivatives cannot be meaningfully visualized within the ALE plot itself, we suggest to visualize it in combination with a plot for the interaction-related heterogeneity measured by the standard deviation of partial derivatives within each interval, which is inspired by the derivative ICE plots of Goldstein et al. (2015) (see e.g., Figure 12).

DECOMPOSING GLOBAL FEATURE EFFECTS

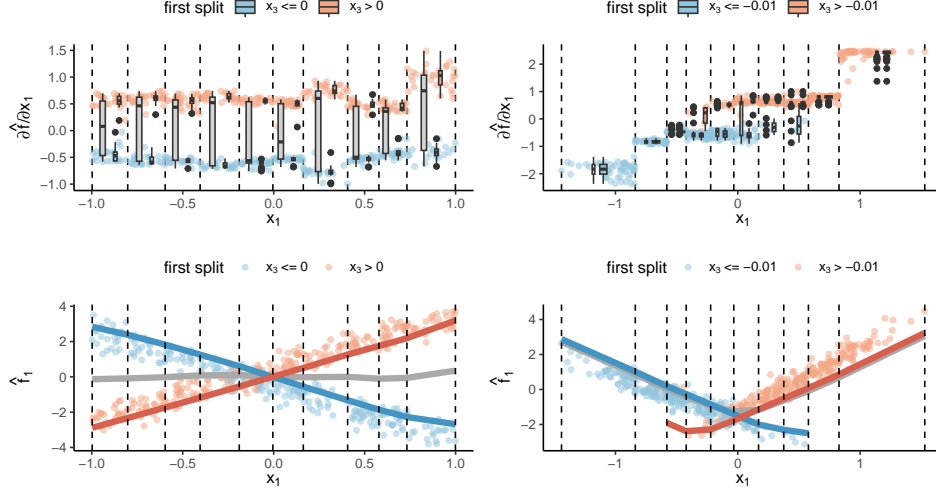


Figure 4: Visualization of derivatives of \hat{f} w.r.t. \mathbf{x}_1 (top) and respective ALE curves (bottom) for the uncorrelated (left) and correlated case (right) of the simulation example of Section 3, with $Y = 3X_1 \mathbb{1}_{X_3 > 0} - 3X_1 \mathbb{1}_{X_3 \leq 0} + X_3 + \epsilon$ and $\epsilon \sim \mathcal{N}(0, 0.09)$. Derivatives (top) and mean-centered observational values (bottom) are colored w.r.t. the first split when GADGET-ALE with $S = 1$ and $Z = \{2, 3\}$ is applied. The boxplots (top) and lines (bottom) in grey show the variation of derivatives and the global centered ALE curves, respectively. The colored curves show the regional centered ALE curves after the first split with GADGET-ALE.

4.5 GADGET-SD

While PD and ALE plots visualize the global feature effect for which we define the appropriate local feature effect function (mean-centered ICE curves and derivatives) for the GADGET algorithm, the SD plot (Section 2.3) is comparable to an ICE plot and, thus, does not show the global feature effect itself. Here, we provide an estimate for the global effect and show the applicability of SD within GADGET which we term GADGET-SD.

Method. Herren and Hahn (2022) (amongst others) showed that the Shapley value $\phi_j^{(i)}(x_j)$ of observation i for feature value $x_j = \mathbf{x}_j^{(i)}$ can be decomposed as defined in Eq. (8) to

$$\phi_j^{(i)}(x_j) = g_j^c(x_j) + \sum_{k=1}^{p-1} \frac{1}{k+1} \sum_{W \subseteq -j; |W|=k} g_{W \cup j}^c(x_j, \mathbf{x}_W^{(i)}),$$

with $g_{W \cup j}^c(x_j, \mathbf{x}_W^{(i)}) = \mathbb{E}[\hat{f}(x_j, X_{-j}) | X_W = \mathbf{x}_W^{(i)}] - \sum_{V \subset \{W \cup j\}} \mathbb{E}[\hat{f}(X) | X_V = \mathbf{x}_V^{(i)}]$, which satisfies Axiom 1 (see Appendix B.6). The global feature effect for the SD plot of feature

\mathbf{x}_j at the feature value $x_j = \mathbf{x}_j^{(i)}$ can then be defined by

$$f_j^{SD}(x_j) = \mathbb{E}_{X_W}[\phi_j(x_j)] = g_j^c(x_j) + \sum_{k=1}^{p-1} \frac{1}{k+1} \sum_{W \subseteq -j: |W|=k} \mathbb{E}[g_{W \cup j}^c(x_j, X_W)]. \quad (16)$$

Following from that, according to Theorem 2, the respective loss function used in GADGET depends only on interaction effects between \mathbf{x}_j and features in $-j$ and is given by

$$\mathcal{L}^{SD}(\mathcal{A}_g, x_j) = \sum_{i: \mathbf{x}^{(i)} \in \mathcal{A}_g \wedge \mathbf{x}_j^{(i)} = x_j} \left(\phi_j^{(i)}(x_j) - \mathbb{E}_{X_W}[\phi_j(x_j) | \mathcal{A}_g] \right)^2. \quad (17)$$

For more details, see Appendix B.

While there exist estimators for the global effect in PD and ALE plots, a pendant for the SD plot has not been introduced yet. Hence, a suitable estimator to estimate the expected value of Shapley values of feature \mathbf{x}_j in Eq. (17) must be chosen. Here, we use univariate GAMs with splines to estimate the expected value.¹⁵ Thus, the estimated GAM for feature \mathbf{x}_j represents the regional SD feature effect of feature \mathbf{x}_j within subspace \mathcal{A}_g .

Illustration. Figure 5 visualizes the SD plot for feature \mathbf{x}_1 of the simulation example described in Section 3. Similarly to the least-square estimate in linear regression, we search for the GAM that minimizes the squared distance (Δ^2) of the Shapley values of \mathbf{x}_1 . With GADGET, we now split such that the fitted GAMs within the two new subspaces minimize the squared distances between them and the Shapley values within the respective subspace. Since the GAMs are fitted on the Shapley values (local feature effects), in contrast to PDs, they do not extrapolate with regard to \mathbf{x}_1 in the correlated scenario. However, as defined in the beginning of this section, Shapley values are based on expected values that must be estimated. If they are calculated using the interventional approach (as we do here), it is still possible that the predictions considered in the Shapley values extrapolate in sparse regions. Hence, the definition of Shapley values via expected values differs from those of ICE curves and derivatives for ALE, which are based on local predictions. Similarly to estimating the mean-centering constants for ICE curves, it follows that the expected values within the Shapley value estimation must consider the regarded subspace \mathcal{A}_g to acknowledge the full heterogeneity reduction due to interactions within each subspace. This means that we must recalculate the Shapley values after each partitioning step for each new subspace to receive regional SD effects in the final subspaces that are representative of the underlying main effects within each subspace. However, without recalculating the conditional expected values, we still minimize the unconditional expected value (i.e., the feature interactions on the entire feature space). The two different approaches lead to a different acknowledgment of interaction effects. In general, it can be said that the faster approach without recalculation will be less likely to detect feature interactions of higher order compared to the exact approach with recalculation. The differences of the two approaches for the simulation example of Section 3 are explained in Appendix C.2 and further discussed on a more complex simulation example in Section 6.2.

15. Splines are functions that are defined in a piece-wise manner by polynomials. Splines are often preferred over polynomial regression, since they provide more flexibility with already low-order polynomials.

DECOMPOSING GLOBAL FEATURE EFFECTS

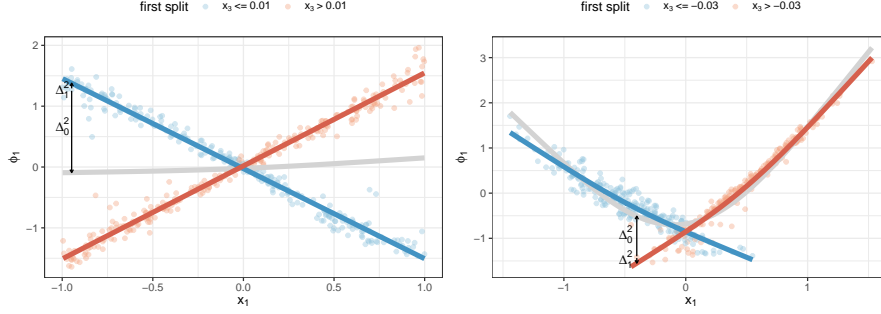


Figure 5: Visualization of Shapley values ϕ_1 w.r.t. x_1 for the uncorrelated (left) and correlated case (right) of the simulation example of Section 3, with $Y = 3X_1 \mathbb{1}_{X_3 > 0} - 3X_1 \mathbb{1}_{X_3 \leq 0} + X_3 + \epsilon$ and $\epsilon \sim \mathbb{N}(0, 0.09)$. Shapley values (points) and the respective regional feature effect curves (GAMs) are colored w.r.t. the first split when GADGET-SD with $S = 1$ and $Z = \{2, 3\}$ is applied. The grey curve represents the global feature effect based on the entire feature space. Δ_0^2 represents the squared distance between a given Shapley value and the global SD curve (grey), while Δ_1^2 measures the squared distance between this Shapley value and the respective regional SD curve. The GAMs are fitted such that these squared distances over all Shapley values in the respective regions are minimized.

Decomposition. The decomposition of Shapley values differs from that of mean-centered ICE curves in the way that feature interactions in Shapley values are fairly distributed between involved features, which leads to decreasing weights the higher the order of the interaction effect (see Eq. 8). In contrast, all interaction terms receive the same weight as the main effects in ICE curves. Hence, interactions of high order lead to less heterogeneity in SD plots compared to ICE plots. However, by using the interventional approach to calculate Shapley values, they can be decomposed by weighted PD functions (see Eq. 8 and Herren and Hahn, 2022). Hence, the same decomposition rules as defined for PD plots apply for the SD feature effect, as defined in Eq. (16). Meaning, if Z contains all features interacting with features in S , and if GADGET is applied such that the theoretical minimum of the objective function is reached, then according to Corollary 4, the regional joint SD effect of features in S can be decomposed into 1-dimensional regional SD effect functions, as in Eq. (12). In this special case and if Shapley values are estimated by the interventional approach, it can be shown that $f_{j|\mathcal{A}_g}^{SD,c} = f_{j|\mathcal{A}_g}^{PD,c}$ (see Appendix C.2).

Again, we might be able to decompose the prediction function into the regional features' main effects, depending on how we choose the feature subsets S and Z within the GADGET algorithm. More details and an exemplary illustration of the uncorrelated simulation example of Section 3 can be found in Appendix C.2.

Estimates and Visualization. The regional SD effect is estimated by a GAM for each feature in S and final region \mathcal{A}_g . The interaction-related heterogeneity for each feature in

S and final subspace \mathcal{A}_g is measured by the risk function in Eq. (11) with the loss function in Eq. (17). The risk function quantifies the variation of Shapley values of feature \mathbf{x}_j within each region. We visualize the interaction-related heterogeneity by the Shapley values that are recalculated conditioned on the subspace \mathcal{A}_g (see, e.g., Figure 18 in Appendix E).

An overview of introduced estimates and visualization techniques for the different effect methods can be found in Appendix C.3. Moreover, an explanation of how categorical features are handled within GADGET for each of the presented feature effect methods is provided in Appendix C.4.

4.6 Quantifying Feature Interactions

Since GADGET minimizes the interaction-related heterogeneity based on the underlying feature effect method, we can quantify feature interactions by measuring the heterogeneity reduction in each partitioning step. We introduce several measures (inspired by Herbinger et al., 2022) to gain more insights into learned feature interactions and the meaningfulness of the final regional effects based on the interaction-related heterogeneity reduction.

Analysis of Single Partitioning Steps. We can quantify the extent to which interaction-related heterogeneity has been reduced in each of the features in S after splitting according to the chosen split feature $z \in Z$. Here, we denote G to be the total number of nodes after applying GADGET. Furthermore, we denote P to be the index of a parent node and the indices l and r to be the respective child nodes. We can quantify the relative interaction-related heterogeneity reduction after the respective partitioning step for feature $j \in S$ by

$$I(\mathcal{A}_P, \mathbf{x}_j) = \frac{\mathcal{R}(\mathcal{A}_P, \mathbf{x}_j) - \mathcal{R}(\mathcal{A}_l, \mathbf{x}_j) - \mathcal{R}(\mathcal{A}_r, \mathbf{x}_j)}{\mathcal{R}(\mathcal{X}, \mathbf{x}_j)},$$

meaning that we quantify the risk reduction of the regarded split relative to the risk on the entire feature space (root node) for one feature of interest $j \in S$. For example, in Figure 3, the interaction-related heterogeneity reduction of the first split ($I(\mathcal{X}, \mathbf{x}_1)$) for feature \mathbf{x}_1 is 0.986, which means that almost all of the interaction-related heterogeneity of \mathbf{x}_1 is reduced after the first split.

Analysis of Single Split Features. Instead of considering only a single partitioning step, one might be more interested in how much interaction-related heterogeneity reduction a specific split feature $z \in Z$ is responsible for w.r.t. all performed partitioning steps of an entire tree. To that end, we can quantify (either for each feature of interest in S or for all features in S together) the relative amount of heterogeneity reduction due to feature z . Hence, we receive $I_{z,j}(\mathbf{x}_j)$ by summing up $I(\mathcal{A}_P, \mathbf{x}_j)$ for the set of parent subspaces that used feature z as split feature and that we denote by $\mathcal{B}_z \subset \{\mathcal{A}_1, \dots, \mathcal{A}_G\}$:

$$I_{z,j}(\mathbf{x}_j) = \sum_{\mathcal{A}_P \in \mathcal{B}_z} I(\mathcal{A}_P, \mathbf{x}_j).$$

We obtain the overall interaction-related heterogeneity reduction I_z of the z -th split feature by first aggregating over all features in S :

$$I_z = \frac{\sum_{\mathcal{A}_P \in \mathcal{B}_z} \sum_{j \in S} (\mathcal{R}(\mathcal{A}_P, \mathbf{x}_j) - \mathcal{R}(\mathcal{A}_l, \mathbf{x}_j) - \mathcal{R}(\mathcal{A}_r, \mathbf{x}_j))}{\sum_{j \in S} \mathcal{R}(\mathcal{X}, \mathbf{x}_j)}.$$

DECOMPOSING GLOBAL FEATURE EFFECTS

In our previous example, we obtain $I_{3,1}(\mathbf{x}_1) = I(\mathcal{X}, \mathbf{x}_1) = 0.986$, since $z = 3$ was only used once for splitting, while the interaction-related heterogeneity reduction of all three features is $I_3 = 0.99$ for $z = 3$.

Goodness of Fit. A further aggregation level would be to sum up $I_{z,j}(\mathbf{x}_j)$ over all $z \in Z$ and, thus, receive the interaction-related heterogeneity reduction for feature \mathbf{x}_j between the entire feature space and the final subspaces (hence, over the entire tree). This is related to the concept of R^2 , which is a well-known measure in statistics to quantify the goodness of fit. We apply this concept here to quantify how well the final regional effect curves (in the final subspaces $\mathcal{B}_t \subset \{\mathcal{A}_1, \dots, \mathcal{A}_G\}$) fit the underlying local effects compared to the global feature effect curve on the entire feature space. We distinguish between the feature-related R_j^2 , which represents the goodness of fit for the feature effects of feature \mathbf{x}_j :

$$R_j^2 = \sum_{z \in Z} I_{z,j}(\mathbf{x}_j) = 1 - \frac{\sum_{\mathcal{A}_t \in \mathcal{B}_t} \mathcal{R}(\mathcal{A}_t, \mathbf{x}_j)}{\mathcal{R}(\mathcal{X}, \mathbf{x}_j)},$$

and the R_{Tot}^2 , which quantifies the goodness of fit for the feature effects of all features in S :

$$R_{Tot}^2 = \sum_{z \in Z} I_z = 1 - \frac{\sum_{\mathcal{A}_t \in \mathcal{B}_t} \sum_{j \in S} \mathcal{R}(\mathcal{A}_t, \mathbf{x}_j)}{\sum_{j \in S} \mathcal{R}(\mathcal{X}, \mathbf{x}_j)}.$$

Both R^2 measures take values between 0 and 1, with values close to 1 signalling that almost all heterogeneity in the final subspaces compared to the entire feature space has been reduced—either for a specific feature of interest (R_j^2) or for all features of interest (R_{Tot}^2). In our example, R_1^2 for feature \mathbf{x}_1 is the same as for $I_{3,1}(\mathbf{x}_1)$, since GADGET performed only one split. The total interaction-related heterogeneity reduction over all features in S is $R_{Tot}^2 = I_3 = 0.99$. Hence, the interaction-related heterogeneity of all features in S has been reduced by 99% after the first split.

These measures provide a tool set to better understand how features interact with each other and how well the final regional effect plots represent the underlying local effects.

4.7 Choosing Stop Criteria

The question of how many partitioning steps should be performed depends on the underlying research question. If the user is more interested in reducing the interaction-related heterogeneity as much as possible, they might split rather deeply, depending on the complexity of interactions learned by the model. However, this might lead to many regions that are more challenging to interpret. If the user is more interested in a small number of regions, they might prefer a shallow tree, thus reducing only the heterogeneity of the features that interact the most.

Here, we suggest the following stopping criteria to control the number of partitioning steps in GADGET: First, we could choose common hyperparameters of a decision tree, like the tree depth or the minimum number of observations per leaf node. Another option is to apply an early stop mechanism based on the interaction-related heterogeneity reduction—either in each split or in total. According to our proposed split-wise measure, a further split is only performed if the relative improvement of the split to be performed is at least

$\gamma \in [0, 1]$ times the total relative interaction-related heterogeneity reduction of the previous split: $\gamma \times \frac{\sum_{j \in S} (\mathcal{R}(A_P, \mathbf{x}_j)) - \mathcal{I}(i, \hat{z})}{\sum_{j \in S} (\mathcal{R}(\mathcal{X}, \mathbf{x}_j))}$. Another possibility is to stop splitting as soon as a pre-defined total reduction of heterogeneity (R_{Tot}^2) is reached. In general, it holds that the higher we choose γ and the lower we choose the threshold for R_{Tot}^2 , the fewer partitioning steps will be performed, and vice versa.

5 Significance Test for Global Feature Interactions

In addition to the hyperparameters for early stopping, there are two more hyperparameters in Algorithm 1 that must be specified—namely, the features of interest contained in S and the interacting (splitting) feature subset Z . Choosing the features to be contained in S and Z strongly depends on the underlying research question. If the user is interested in how a specific set of features (S) influences the model’s predictions depending on another user-defined feature set (Z), then S and Z are chosen based on domain knowledge. However, the user does not know which feature effects and interactions were inherently learned by the ML model. Thus, choosing S and Z based on domain knowledge does not guarantee that all interacting features are considered and that we can additively decompose the prediction function into univariate feature effects.

Thus, if our goal is to minimize feature interactions between all features to additively decompose the prediction function into mainly univariate effects, we can define $S = Z = \{1, \dots, p\}$. With that choice, we aim to reduce the overall interaction-related heterogeneity in all features (S), since we also consider all features to be possible interacting features (Z). However, this choice has two disadvantages. First, we must loop over all features and possible split points in each partitioning step, which may be slow in medium- or high-dimensional settings. This might also lead to less stable results, since only a few features might interact with each other, although many more features are considered for splitting. Second, if features are correlated, we might obtain spurious interactions (which we do not want to consider, since we are only interested in true interaction effects). Hence, to solely split according to feature interactions and only measure the interaction-related heterogeneity reduction, we must define beforehand the subset of features that actually interact. Since features interact with each other and all involved features will usually show heterogeneity in their local effects while being responsible for the heterogeneity of other involved features, we will usually choose $S = Z$.¹⁶ Thus, we will furthermore only use S as the globally interacting feature subset to be defined.

Since the H-Statistic (as defined in Section 2.4) is a global interaction measure, it could be used to define S by choosing all features with a high H-Statistic value. However, the question remains of which value is considered “high”. Furthermore, the H-Statistic is based on PDs. Thus, the interacting features are always chosen depending on the interaction quantification of PDs, which might differ from other feature effect methods (e.g., see the discussion of Shapley values versus ICE curves in Section 4.5). Since it is based on PDs, the H-Statistic might also suffer from detecting spurious interactions (Sorokina et al., 2008).

We introduce a new statistical permutation interaction test (PINT) that is inspired by the permutation importance (PIMP) algorithm of Altmann et al. (2010) to test for

¹⁶. Z might differ in the case of non-symmetrical interactions (see Section 7).

DECOMPOSING GLOBAL FEATURE EFFECTS

significant feature importance values. The goal of PINT (see Algorithm 2) is to define the feature subset S that contains all features that significantly interact with each other, w.r.t. a predefined significance level α and a chosen feature effect method. With the risk function defined in Eq. (11) and based on the chosen local feature effect method, we can quantify the interaction-related heterogeneity of each feature \mathbf{x}_j within the feature space \mathcal{X} . Due to correlations between features, the estimated heterogeneity might also include spurious interactions. Thus, we must define a null distribution to determine which heterogeneity is actually due to feature interactions (i.e., significant w.r.t. the null distribution) and which heterogeneity is due to other reasons, such as correlations or noise. This is achieved by permuting the target variable y (line 4 in Algorithm 2). With that, we break the association between the features and the target variable, but the underlying data structure remains. We refit the given ML model based on the data set $\tilde{\mathcal{D}}$ with the permuted target variable and calculate the respective risk $\tilde{\mathcal{R}}_j$ for each feature \mathbf{x}_j based on the chosen feature effect method h (lines 5-8 in Algorithm 2). This is repeated s times, producing s permuted risk values that represent the null distribution for the unpermuted risk value $\mathcal{R}(\mathcal{X}, \mathbf{x}_j)$ of the j -th feature. Then, we perform a statistical test based on the null hypothesis $H_0 : \mathcal{R}(\mathcal{X}, \mathbf{x}_j) \leq \tilde{\mathcal{R}}_j^{(s \cdot (1-\alpha))}$. Hence, if the unpermuted risk value is larger than the $(1-\alpha)$ -quantile of the null distribution, then the j -th feature is significant w.r.t. the defined α -level and belongs to the interacting feature subset S (lines 11-18 in Algorithm 2).

Algorithm 2: PINT

```

1: input: data set  $\mathcal{D}$ , prediction function  $\hat{f}$ , number of permutations  $s$ ,
   risk function  $\mathcal{R}$ , significance level  $\alpha$ 
2: output: feature subset  $S$ 
3: for  $k \in \{1, \dots, s\}$  do
4:   permute  $y$  of  $\mathcal{D}$  denoted by  $\tilde{y}^k$  and  $\tilde{\mathcal{D}}^k = \{\mathbf{x}, \tilde{y}^k\}$ 
5:   refit model on  $\tilde{\mathcal{D}}^k$  to obtain the prediction function  $\tilde{f}^k$ 
6:   for  $j \in \{1, \dots, p\}$  do
7:     calculate risk  $\tilde{\mathcal{R}}_j^{(k)} = \tilde{\mathcal{R}}^k(\mathcal{X}, \mathbf{x}_j)$  for  $j$ -th feature based on  $\tilde{\mathcal{D}}^k$  and  $\tilde{f}^k$ 
8:   end for
9: end for
10: for  $j \in \{1, \dots, p\}$  do
11:   a) calculate risk  $\mathcal{R}(\mathcal{X}, \mathbf{x}_j)$  for  $j$ -th feature based on  $\mathcal{D}$  and  $\hat{f}$ 
12:   b) sort  $\tilde{\mathcal{R}}_j$  in increasing order
13:   c) determine the  $(1 - \alpha)$ -quantile of permuted risk values  $z_j^{1-\alpha} = \tilde{\mathcal{R}}_j^{(s \cdot (1-\alpha))}$ 
14:   if  $\mathcal{R}(\mathcal{X}, \mathbf{x}_j) > z_j^{1-\alpha}$  then
15:      $j \in S$ 
16:   else
17:      $j \notin S$ 
18:   end if
19: end for
    
```

We illustrate the performance of PINT compared to the H-Statistic on an example which might be affected by spurious interactions. Therefore, we consider that $X_1, X_2, X_4 \sim$

$U(-1,1)$ and $X_3 = X_2 + \epsilon$ with $\epsilon \sim N(0,0.09)$. We draw $n = \{300, 500\}$ observations of these four random variables and assume the following relationship between \mathbf{y} and \mathbf{x} : $\mathbf{y} = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3 - 2\mathbf{x}_1\mathbf{x}_2$. Hence, \mathbf{x}_1 and \mathbf{x}_2 have a negative linear interaction effect, while \mathbf{x}_2 and \mathbf{x}_3 are highly linearly positively correlated but do not interact. Thus, there might be a spurious interaction between \mathbf{x}_1 and \mathbf{x}_3 . We calculate PINT and the H-Statistic for all features using a support vector machine (SVM) as the underlying ML model (with specifications defined in Section 6.3). We repeat the experiment 30 times. Figure 8 shows that for \mathbf{x}_1 and \mathbf{x}_2 , the PINT test is significant for almost all repetitions and effect methods, while it is never significant for features \mathbf{x}_3 and \mathbf{x}_4 w.r.t. a significance level of $\alpha = 0.05$. Hence, besides a few exceptions for ALE, the PINT algorithm returns the correct interacting feature subset $S = \{1, 2\}$. The H-Statistic shows that \mathbf{x}_1 interacts most with all other features, then \mathbf{x}_2 , followed by \mathbf{x}_3 and \mathbf{x}_4 in the rankings. Depending on which threshold is chosen, one would possibly include the non-interacting feature \mathbf{x}_3 in S , which shows over all repetitions values ranging from 0.1 to 0.2—possibly due to spurious interactions.

Hence, PINT not only allows to more clearly and (in this example) correctly define the subset S than the H-Statistic, but PINT also has the advantage that it can be used with any feature effect method that we use for GADGET. Thus, PINT can be applied according to the objective we want to minimize. This is analyzed in more detail in Section 6.

PINT also entails two hyperparameters α and s that must be specified. The significance level α can be chosen depending on the underlying research question. If we are only interested in a small set of very strong interactions, we choose α to be very small. If we want to find all (also small) interaction effects, we choose α to be larger. However, a larger significance level might also lead to detecting spurious interactions. The number of permutations s should be chosen to be as high as possible in order to obtain accurate results. However, since PINT must refit the model within each permutation, the computational burden increases with more permutations. One possible solution to address this trade-off for PIMP was proposed by Altmann et al. (2010), where the authors use a smaller number of permutations (e.g., 100) to approximate the empirical null distribution and then fit a theoretical distribution (e.g., normal, log-normal or gamma distribution) on the empirical distribution. Based on a Kolmogorov-Smirnov test, the theoretical distribution that best fits the empirical distribution is selected to approximate the null distribution. If the Kolmogorov-Smirnov test is not significant for any of the theoretical distributions, then the empirical distribution is used as the null distribution.¹⁷ This approach can similarly be applied for PINT. More suggestions to decrease the computational burden of PINT are provided in Appendix D.

6 Simulations

In this section, we analyze different hypotheses to (1) empirically validate that GADGET generally minimizes feature interactions and (2) show how different characteristics of the underlying data—such as correlations between features and different settings of the data-generating process—might influence GADGET, depending on the chosen feature effect method. The structure of the following sections and the concrete definition of the hypotheses is based on (2). However, the simulation examples themselves are designed in such a way

17. Note that Altmann et al. (2010) do not adjust for multiple testing, which can lead to false positives in higher dimensions (Molnar et al., 2022).

that we know the underlying ground-truth of feature interactions for each example. Thus, we are able to empirically validate that GADGET generally minimizes feature interactions.

6.1 Extrapolation

Hypothesis. For increasing correlation between features, due to extrapolation, we receive results that are less stable for methods using the marginal distribution for integration (especially PD, as shown in Section 3, but also SD) compared to methods using the conditional distribution (such as ALE). Note that in this context, “stability” refers to the ability of the different methods to find the correct split feature and split point in GADGET over various repetitions.

Experimental Setting. To address this hypothesis, we choose the (simple) simulation example of Section 3 and compare GADGET based on PD, ALE, and SD for four different correlation coefficients ρ_{13} between \mathbf{x}_1 and \mathbf{x}_3 —0, 0.4, 0.7, and 0.9. The data is generated as follows: Let $X_2, X_3 \sim \mathcal{U}(-1, 1)$ be independently distributed and $X_1 = c \cdot X_3 + (1 - c) \cdot Z$ with $Z \sim \mathcal{U}(-1, 1)$, where c takes values between 0 and 0.7, which correspond to the above-mentioned ρ_{13} values. The true underlying relationship is defined as before by $Y = 3X_1\mathbb{1}_{X_3 > 0} - 3X_1\mathbb{1}_{X_3 \leq 0} + X_3 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.09)$. We draw 1000 observations and fit a GAM with correctly specified main and interaction effects as well as an NN with the previously defined specifications to the data. We repeat the experiment 30 times.

We apply GADGET to each setting and model within each repetition using PD, ALE, and SD as feature effect methods. We consider all features as features of interest as well as potential interacting (split) features, meaning $S = Z = \{1, 2, 3\}$. As stopping criteria, we choose a maximum depth of 6, minimum number of observations per leaf of 40, and set the improvement parameter γ to 0.2.

Results. Figure 6 shows that independent of the model or correlation degree, \mathbf{x}_2 has (correctly) never been considered as split feature. For correlation strengths ρ_{13} between 0 and 0.7, \mathbf{x}_3 is always chosen as the only split feature, with I_3 taking values between 0.75 and 1 and, thus, reducing most of the interaction-related heterogeneity of all features with one split. Thus, for low to medium correlations, there are only minor differences between the various feature effect methods and models. However, the observed behavior changes substantially for $\rho_{13} = 0.9$. For the correctly-specified GAM, we still receive quite consistent results, apart from one repetition where \mathbf{x}_1 is chosen as the split feature when PD is used. In contrast, there is more variation of I_3 when the NN is considered as the underlying ML model, especially when SD is used. For SD, \mathbf{x}_1 is chosen once for splitting, and for PD, this is also the case for 30% of all repetitions (see Table 1). While using ALE, GADGET always correctly performs one split with regard to \mathbf{x}_3 . Additionally, GADGET performs a second split once when PD is used and 7 times when SD is used.

Thus, for a high correlation between features, we already observe in this very simple setting that the extrapolation problem influences the splitting within GADGET for effect methods based on marginal distributions, while methods based on conditional distributions such as ALE are less affected. This is particularly relevant for learners that model very locally (e.g., NNs) and thus, tend to have wiggly prediction functions and oscillate in ex-

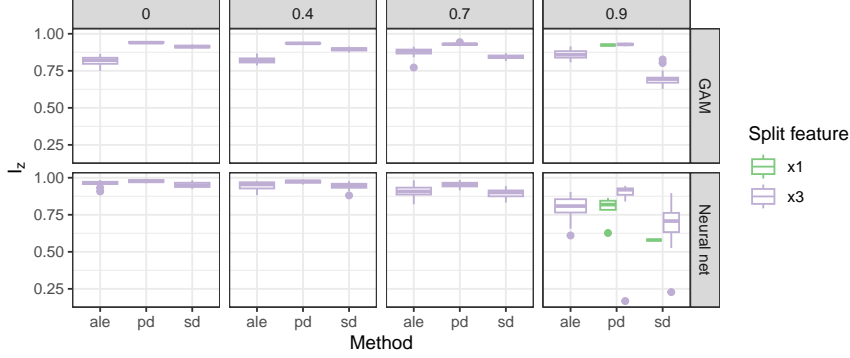


Figure 6: Boxplots showing the interaction-related heterogeneity reduction I_z per split feature over 30 repetitions when PD, ALE, or SD is used in GADGET. The columns show the results depending on the correlation between \mathbf{x}_1 and \mathbf{x}_3 , while the rows show the respective boxplots when GADGET is applied, based on the predictions of a correctly-specified GAM (upper) and of an NN (lower).

trapolating regions. This is also notable on the split value range, which increases for the NN in the case of high correlations for PD and SD but not for ALE (see Table 1).

Model	ρ_{13}	Split feature \mathbf{x}_3 in %			Split value range			MSE
		ALE	PD	SD	ALE	PD	SD	mean (sd)
GAM	0	1.00	1.00	1.00	0.10	0.10	0.12	0.329 (0.008)
GAM	0.4	1.00	1.00	1.00	0.13	0.10	0.10	0.201 (0.003)
GAM	0.7	1.00	1.00	1.00	0.09	0.10	0.09	0.137 (0.002)
GAM	0.9	1.00	0.97	1.00	0.11	0.11	0.12	0.107 (0.001)
NN	0	1.00	1.00	1.00	0.10	0.09	0.09	0.152 (0.018)
NN	0.4	1.00	1.00	1.00	0.08	0.08	0.07	0.124 (0.007)
NN	0.7	1.00	1.00	1.00	0.11	0.10	0.10	0.111 (0.005)
NN	0.9	1.00	0.70	0.97	0.10	0.19	0.15	0.104 (0.003)

Table 1: Overview of how often \mathbf{x}_3 was chosen as the first split feature over all 30 repetitions, including the respective split value range. The last column shows the test performance of the respective model by the mean (standard deviation) of the mean squared error (MSE).

6.2 Higher-Order Effects

Hypothesis. While SD puts less weight on interactions with increasing order, all interactions (independent of the order) receive the same weight in PD and ALE. Hence, when

DECOMPOSING GLOBAL FEATURE EFFECTS

GADGET-SD is used, we might not be able to detect interactions of high order—especially when the approximation without recalculation after each split is used (see Section 4.5). However, it should be more likely to detect these interactions with PD and ALE.

Experimental Setting. To investigate this hypothesis, we consider 5 features with $X_1 \sim \mathbb{U}(0, 1)$ and $X_2, X_3, X_4, X_5 \sim \mathbb{U}(-1, 1)$ and draw 1000 samples. The data-generating process is defined by a series of interactions between different features: $y = f(\mathbf{x}) + \epsilon$ with $f(\mathbf{x}) = \mathbf{x}_1 \cdot \mathbb{1}_{\mathbf{x}_3 \leq 0} \mathbb{1}_{\mathbf{x}_4 > 0} + 4\mathbf{x}_1 \cdot \mathbb{1}_{\mathbf{x}_3 \leq 0} \mathbb{1}_{\mathbf{x}_4 \leq 0} - \mathbf{x}_1 \cdot \mathbb{1}_{\mathbf{x}_3 > 0} \mathbb{1}_{\mathbf{x}_5 \leq 0} \mathbb{1}_{\mathbf{x}_2 > 0} - 3\mathbf{x}_1 \cdot \mathbb{1}_{\mathbf{x}_3 > 0} \mathbb{1}_{\mathbf{x}_5 \leq 0} \mathbb{1}_{\mathbf{x}_2 \leq 0} - 5\mathbf{x}_1 \cdot \mathbb{1}_{\mathbf{x}_3 > 0} \mathbb{1}_{\mathbf{x}_5 > 0}$ and $\epsilon \sim \mathcal{N}(0, 0.01 \cdot \sigma^2(f(\mathbf{x})))$. These interactions can be seen as one hierarchical structure between all features, where the slope of \mathbf{x}_1 depends on the subspace defined by the interacting features (see Figure 7).

We fit an xgboost (XGB) model with correctly-specified feature interactions and a random forest (RF) on the data set and apply GADGET using PD, ALE, SD with recalculation after each split, and SD without recalculation on each model. We repeat the experiment 30 times, where the XGB showed an average (standard deviation) MSE of 0.068(0.009) and the RF of 0.121(0.012) on a separate test set of the same distribution. For GADGET, we consider one feature of interest $S = 1$ and all other features as potential interacting features $Z = \{2, 3, 4, 5\}$. As stop criteria, we choose a maximum tree depth of 7, minimum number of observations of 40, and $\gamma = 0.1$. We can assume that if the underlying model learned the effects of the data-generating process correctly, GADGET must split as shown in Figure 7 to maximally reduce the interaction-related heterogeneity of \mathbf{x}_1 .

Results. For the first partitioning step, all methods used \mathbf{x}_3 as the first split feature in all repetitions. Table 2 shows that a second-level split was performed in only 10% of the repetitions for XGB when SD without recalculation is used within GADGET, while the second-level split frequency for all other methods is approximately 90%. A similar trend is observable when RF is used as underlying ML model but with higher variation in the relative frequencies, which might be due to different learned effects. If further splits for the second depth of the tree are performed, the correct split features \mathbf{x}_4 and \mathbf{x}_5 are always chosen by all methods, which shows that GADGET generally minimizes feature interactions (see Figure 7). Table 3 shows the same difference in relative frequencies between SD without recalculation and all other methods for the third-level splits as for the previous splits. This is confirmed by Table 4, which shows that while SD without recalculation demonstrates a high variation of final subspaces, the median number of subspaces is 2, indicating that this method stops after the first split (with \mathbf{x}_3). The other methods show a higher number of final subspaces; PD and SD with recalculation typically show the correct final number of subspaces (i.e., 5), while ALE tends to split slightly deeper.

To summarize, when SD without recalculation is used, the two-way interaction with \mathbf{x}_3 is primarily detected, while features of a third (\mathbf{x}_4 and \mathbf{x}_5) or fourth order (\mathbf{x}_2) interaction are rarely considered for splitting. By contrast, for most repetitions of the other three methods, we find that interactions of a higher order are also detected. This supports our hypothesis regarding higher-order effects and the theoretical differences of the considered feature effect methods.

Note that due to recalculating the Shapley values after each split, the order of interactions is reduced. For example, recalculating Shapley values in the subspace $\{\mathcal{X} | \mathbf{x}_3 \leq 0\}$ reduces the three-way interaction $\mathbf{x}_1 \cdot \mathbb{1}_{\mathbf{x}_3 \leq 0} \mathbb{1}_{\mathbf{x}_4 > 0}$ to the two-way interaction $\mathbf{x}_1 \cdot \mathbb{1}_{\mathbf{x}_4 > 0}$ and,

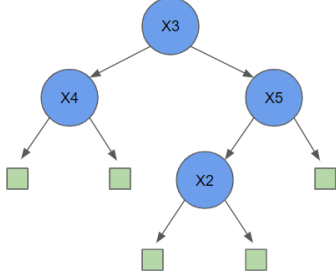


Figure 7: Explanation of the data-generating process. The green squares represent the 5 final subspaces which contain linear effects of feature \mathbf{x}_1 .

Model	Method	$\mathcal{A}_l^{t,z}$		$\mathcal{A}_r^{t,z}$	
		z	Freq.	z	Freq.
XGB	ALE	x4	0.93	x5	0.93
XGB	PD	x4	0.90	x5	0.90
XGB	SD_not_rc	x4	0.10	x5	0.10
XGB	SD_rc	x4	0.87	x5	0.87
RF	ALE	x4	0.80	x5	0.90
RF	PD	x4	0.63	x5	0.73
RF	SD_not_rc	x4	0.03	x5	0.13
RF	SD_rc	x4	0.73	x5	0.90

Table 2: Relative frequencies of splits by split feature z over 30 repetitions in the left $\mathcal{A}_l^{t,z}$ and right $\mathcal{A}_r^{t,z}$ subspaces after the first split when GADGET with ALE, PD, SD with recalculation (rc) and without recalculation (not_rc) is used.

thus, the weight of the interaction increases for the next split. Consequently, we receive similar results for these settings as we do for PD. However, one might consider that the recalculation can be computationally expensive.

Model	Method	z	Rel. Freq.
XGB	ALE	x2	0.93
XGB	PD	x2	0.90
XGB	SD_not_rc	x2	0.10
XGB	SD_rc	x2	0.87
RF	ALE	x2	0.83
RF	ALE	x4	0.03
RF	PD	x2	0.67
RF	SD_not_rc	x2	0.10
RF	SD_rc	x2	0.83

Table 3: Relative frequencies of splits by split feature z over all 30 repetitions in the subspace $\{\mathcal{X} | \mathbf{x}_3 > 0 \cap \mathbf{x}_5 \leq 0\}$ on third tree depth when GADGET is applied with ALE, PD, SD_not_rc and SD_rc.

Model	Method	No. of Subspaces		
		Min	Max	Med
XGB	ALE	3	15	7
XGB	PD	2	7	5
XGB	SD_not_rc	2	7	2
XGB	SD_rc	2	8	5
RF	ALE	3	16	9
RF	PD	2	11	5
RF	SD_not_rc	2	11	2
RF	SD_rc	3	12	5

Table 4: Minimum, maximum and median number of final subspaces over 30 repetitions after GADGET is applied with ALE, PD, SD_not_rc and SD_rc.

6.3 PINT vs. H-Statistic: Spurious Interactions

Hypothesis. When using PINT to pre-select $S = Z \subseteq \{1, \dots, p\}$, we are more likely to actually split according to feature interactions compared to when choosing all features ($S = Z = \{1, \dots, p\}$) or using the H-Statistic values for pre-selection, especially when potential spurious interactions are present.

Experimental Setting. We use the following simulation example described previously in Section 5: We consider four features with $X_1, X_2, X_4 \sim U(-1, 1)$ and $X_3 = X_2 + \epsilon$ with $\epsilon \sim N(0, 0.09)$. For 30 repetitions, we draw $n = \{300, 500\}$ observations and create the dependent variable, including a potential spurious interaction between \mathbf{x}_1 and \mathbf{x}_3 : $\mathbf{y} = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3 - 2\mathbf{x}_1\mathbf{x}_2$. For each sample size n and each repetition, we fit an SVM based on a radial basis function (RBF) kernel with cost parameter $C = 1$ and choose the inverse kernel width based on the data. We receive very similar model performance values—measured by the mean (standard deviation) of the MSE on a separate test set of size 100000 of the same distribution—of 0.028(0.010) and 0.027(0.010) for $n = 300$ and $n = 500$, respectively.

We calculate PINT using PD, ALE, and SD for each repetition and sample size with $s = 100$ and $\alpha = 0.05$ by approximating the null distribution as described in Section 5 and Altmann et al. (2010). We apply GADGET using PD, ALE, and SD with recalculation, where $S = Z$ is based on the feature subset chosen by PINT for the respective feature effect method. We compare these results with considering all features as features of interest and potential split features (i.e. $S = Z = \{1, \dots, p\}$). We use a maximum tree depth of 6, minimum number of observations of 40, and $\gamma = 0.15$ as stop criteria.

Results. The two left plots in Figure 8 show that \mathbf{x}_3 and \mathbf{x}_4 are always correctly identified as insignificant (according to the chosen α level), while the interacting features \mathbf{x}_1 and \mathbf{x}_2 are always significant and thus considered in S (apart from a few exceptions for ALE). The sample size does not seem to have a clear influence on PINT in this setting. The right plots in Figure 8 show that even the H-Statistic values of the non-influential and uncorrelated feature \mathbf{x}_4 are larger than 0 for both sample sizes. The H-Statistic values of \mathbf{x}_3 might support considering \mathbf{x}_3 in S , depending on which threshold is chosen. Since this choice is not clear for the H-Statistic, it is not very suitable as a pre-selection method for GADGET.

Figure 9 illustrates that we correctly only consider \mathbf{x}_1 and \mathbf{x}_2 when PINT is applied upfront, while GADGET also splits w.r.t. \mathbf{x}_3 for all settings and (in some cases) even w.r.t. \mathbf{x}_4 if PINT is not applied upfront. The influence of these two features seems to be higher for the smaller sample size according to I_z . Note that of the various effect methods used in GADGET, PD and SD attribute most of the heterogeneity reduction to \mathbf{x}_1 and a small part to \mathbf{x}_2 , while ALE attributes the heterogeneity more equally between the two split features. A possible explanation might be the correlation between \mathbf{x}_2 and \mathbf{x}_3 , which particularly affects the two methods based on marginal distributions (i.e., PD and SD). Furthermore, Table 5 shows that we tend to obtain shallower trees when we use PINT upfront, while we retain the level of heterogeneity reduction by obtaining similar R_j^2 values for the two interacting features \mathbf{x}_1 and \mathbf{x}_2 .

Thus, PINT reduces the number of features to consider in the interacting feature subset for GADGET depending on the regarded feature effect method. This leads to better results in GADGET in the sense that we only split w.r.t. truly interacting features defined by PINT and receive shallower (and thus, more interpretable) trees.

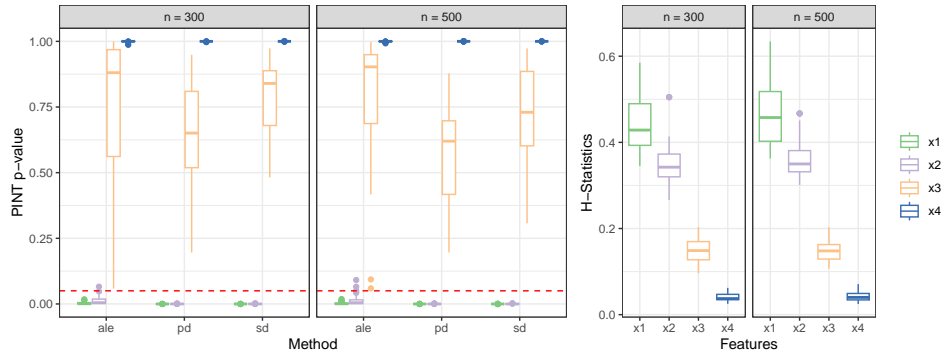


Figure 8: Boxplots showing the distribution of p-values of each feature for different sample sizes and effect methods over all repetitions when PINT is applied (left) and the distribution of feature-wise H-Statistic values for both sample sizes (right).

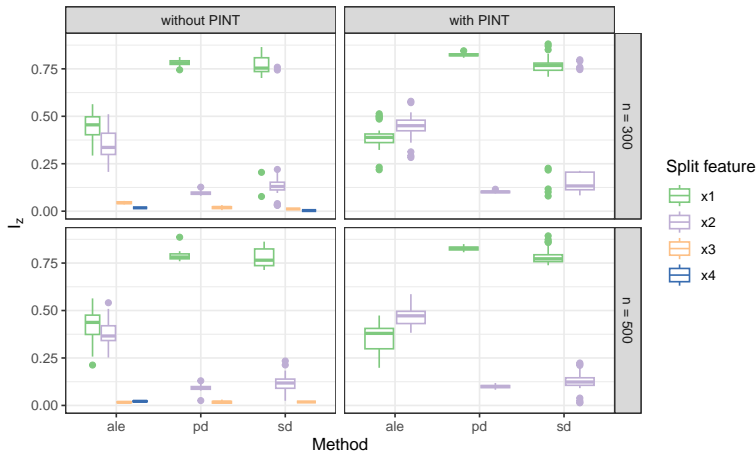


Figure 9: Boxplots showing the interaction-related heterogeneity reduction I_z per split feature over 30 repetitions when PD, ALE, or SD is used within GADGET. The rows show the results for the two different sample sizes, and the columns indicate if GADGET is used based on all features without using PINT upfront (left) or GADGET is used based on the feature subset resulting from PINT (right).

DECOMPOSING GLOBAL FEATURE EFFECTS

n	Method	PINT	R_j^2		Number of Subspaces		
			R_1^2 mean(sd)	R_2^2 mean(sd)	Min	Max	Median
300	ALE	no	0.83 (0.05)	0.83 (0.05)	4	10	7.50
300	ALE	yes	0.83 (0.07)	0.82 (0.05)	1	10	7.00
300	PD	no	0.94 (0.02)	0.84 (0.06)	2	7	4.00
300	PD	yes	0.92 (0.03)	0.80 (0.08)	2	5	3.00
300	SD	no	0.93 (0.04)	0.90 (0.05)	2	11	5.00
300	SD	yes	0.93 (0.04)	0.90 (0.05)	2	11	4.50
500	ALE	no	0.83 (0.04)	0.83 (0.06)	4	10	7.00
500	ALE	yes	0.86 (0.04)	0.82 (0.05)	1	11	6.50
500	PD	no	0.94 (0.03)	0.84 (0.06)	2	7	4.00
500	PD	yes	0.93 (0.03)	0.82 (0.08)	2	5	4.00
500	SD	no	0.93 (0.04)	0.90 (0.05)	2	11	5.00
500	SD	yes	0.93 (0.04)	0.90 (0.05)	2	9	5.00

Table 5: Interaction-related heterogeneity reduction per feature for \mathbf{x}_1 and \mathbf{x}_2 by mean (standard deviation) of R_j^2 and minimum, maximum and median number of final subspaces after applying GADGET based on different sample sizes, effect methods and with and without using PINT upfront.

7 Real-World Applications

In this section, we show the usefulness of our introduced methodology on two real-world application examples. These examples demonstrate that we can both obtain more insights about the learned effects and interactions of the underlying model as well as potentially be able to detect potential bias in the data or model.

COMPAS Data Set. Due to potential subjective judgement and a resulting bias in the decision-making process of the criminal justice system (Blair et al., 2004), ML models have been used to predict recidivism of defendants to provide a more objective guidance for judges. However, if the underlying training data is biased (e.g., different socioeconomic groups have been treated differently for the same crime in the past), the ML model might learn the underlying data bias and, due to its black-box nature, explanations for its decision-making process and a potential recourse are harder to achieve (Fisher et al., 2019).

We want to use GADGET here to obtain more insights in how different characteristics about the defendant and their criminal record influence the risk of recidivism within different subgroups and if “inadmissible” characteristics such as ethnicity or gender cause a different risk evaluation. For our analysis, we use a data set to predict the risk of violent recidivism gathered by ProPublica (Larson et al., 2016) from the Broward County Clerk’s Office, the Broward County Sheriff’s Office, and the Florida Department of Corrections, based on the commercial black-box model of Northpointe Inc. called the Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) data set.

As Fisher et al. (2019), we choose the three admissible features (1) age of the defendant, (2) number of prior crimes, and (3) if the crime is considered a felony versus misdemeanor, and the two “inadmissible” features (1) ethnicity and (2) gender of the defendant. We use

the subset of African-American and Caucasian defendants and apply the data pre-processing steps suggested by ProPublica and applied in Fisher et al. (2019), which leaves us with 3373 defendants of the original pool of 4743 defendants. We consider a binary target variable indicating a high (= 1) or low (= 0) recidivism risk, which is based on a categorization of ProPublica. We perform our analysis on the full data set, using a tuned SVM.¹⁸

Since the features do not show high correlations, we use ICE and PD for our analysis.¹⁹ Figure 10 shows that the average predicted risk visibly decreases with age, while the predicted risk first steeply increases with the number of prior crimes until 10 and then slightly decreases for higher values. The PD values for the type of crime do not differ substantially. When considering the two “inadmissible” features, there is on average a slightly higher risk of recidivism for African-American versus Caucasian and female versus male defendants. For all features, we can observe highly differing local effects. In particular, the heterogeneous ICE curves for age and number of prior crimes indicate potential feature interactions.

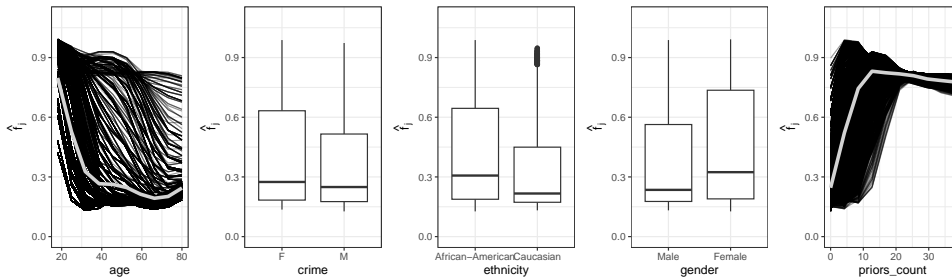


Figure 10: ICE and PD curves of considered features of the COMPAS application example.

We first apply PINT with PD to define our subset S for GADGET. To that end, we choose $s = 200$ and $\alpha = 0.05$ and approximate the null distribution using the procedure described in Section 5. All five features are significant w.r.t. the chosen significance level, and thus we choose $S = Z = \{age, crime, ethnicity, gender, priors_count\}$ for GADGET. We apply GADGET with a maximum depth of 3 and $\gamma = 0.15$. The effect plots for the four resulting regions are shown in Figure 11. GADGET performed the first split according to age and the splits on the second depth according to the number of prior crimes. The total interaction-related heterogeneity reduction is $R_{Tot}^2 = 0.86$. The highest heterogeneity reduction is given by age and number of prior crimes, which interact the most (see Figure 11). For defendants around 20 years old, the predicted risk is generally very high. However, for defendants with a small number of prior crimes, the predicted risk decreases very quickly with increasing age, reaching a low risk and remaining so for people older than 32. In contrast, for defendants with more than four prior crimes, the predicted risk decreases only slowly with increasing age. The regional feature effects of the number of prior crimes show that the interaction-related heterogeneity is small for the subgroup of younger people, while

18. More details on the model selection process can be found in Appendix E.

19. We received similar results by using SD instead of PD within GADGET, see Appendix E.

DECOMPOSING GLOBAL FEATURE EFFECTS

some heterogeneity still remains for defendants older than 32. Additionally, the interaction-related heterogeneity of the three binary features (ethnicity, gender, and crime severity) was reduced, indicating an interaction between each of them and age as well as the number of prior crimes. While the effect on the predicted risk only differs slightly between the categories of the three binary features for older defendants with a lower number of prior crimes and for younger defendants with a high number of prior crimes, greater differences were observed for the other two subgroups. The overall difference in predicted risk for the two inadmissible features of ethnicity and gender seems to be especially high for people older than 32 with a higher number of prior crimes as well as for people younger than 32 with a lower number of prior crimes.

Thus, our analysis has discovered a potentially learned bias regarding ethnicity and gender of the defendant, potentially resulting in more severe predicted risk of recidivism for some defendants than for others. Note that we applied GADGET on an ML model that is fitted on the COMPAS scores and not directly on COMPAS. Consequently, we are not able to draw conclusions about the learned effects of the underlying commercial black-box model. However, GADGET is model-agnostic and can be applied to any accessible black-box model.

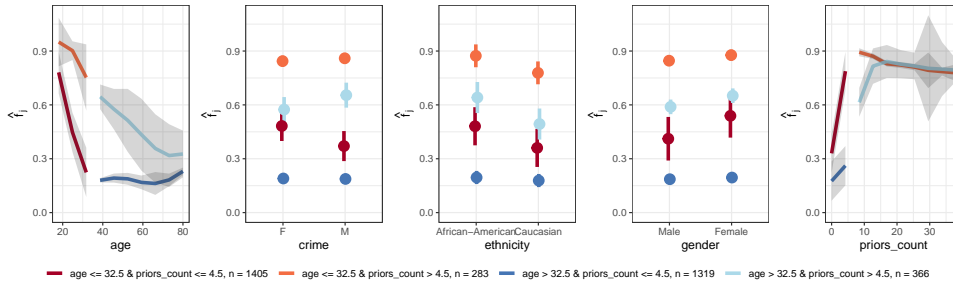


Figure 11: Regional PD plots for all features of the COMPAS application after applying GADGET. The grey areas of the numeric features and the error bars of the categorical features indicate the 95% interaction-related heterogeneity interval as defined in Section 4.3 and Appendix C.3.

Bikesharing Data Set. As a second application example, we choose the bikesharing data set provided in James et al. (2022). The target variable of this regression task is the hourly count of rented bikes for the years 2011 and 2012 in the Capital bikeshare system. The goal here is to predict the hourly rented bikes based on seasonal as well as weather information. We include ten features in our model for this prediction task: the day and hour the bike was rented, if the current day is a working day, the season, and the number of casual bikers. Weather-related features we included are: normalized temperature, perceived temperature, wind speed, humidity and the weather situation (categorical, e.g., clear).

We fit an RF on the data set and use the training data for our further analysis.²⁰

20. More details on the model selection process can be found in Appendix E.

Again, we first apply PINT (as done in the COMPAS example above) with $s = 200$ and $\alpha = 0.05$ on all features to define the interacting feature subset S for GADGET. Since some of the features—such as season, temperature, and perceived temperature—are correlated, we use ALE for our analysis. While the features hour and workingday are highly significant, the p-values of all other features are close to 1, indicating that only the heterogeneity of local effects for hour and workingday are caused by interactions. Hence, we define $S = Z = \{hr, workingday\}$ and apply GADGET with a maximum depth of 3 and $\gamma = 0.15$. GADGET splits once w.r.t. the binary feature workingday, which reduces the interaction-related heterogeneity of the two features by $R_{Tot}^2 = 0.88$. The middle plots of Figure 12 show the regional ALE plots after applying GADGET. High peaks are prominently visible on working days during rush hours, while there is a drop during noon and afternoon hours. However, on non-working days, the trend is the opposite. This interaction is not visible in the global ALE plot of the feature hour (left plot).

The interaction-related heterogeneity of the feature hour is reduced compared to the global plot, although there is still some variation apparent for working days. From a domain perspective, we might also consider an interaction of the temperature with hour and working day (as done in Hiabu et al., 2023). Thus, we include temperature in feature subsets S and Z and apply GADGET again with the same settings as described above. The feature workingday governs the first split. However, for the region of working days, GADGET splits again according to temperature, as shown in the right plot of Figure 12. While the interaction-related heterogeneity of feature temperature was barely reduced within GADGET ($R_j^2 = 0.03$)—which supports the results of PINT—using temperature in the subset of splitting features Z further reduced the interaction-related heterogeneity of hour by 15%. Thus, feature interactions can be asymmetric, and extending Z based on domain knowledge might be a valid option in some cases.

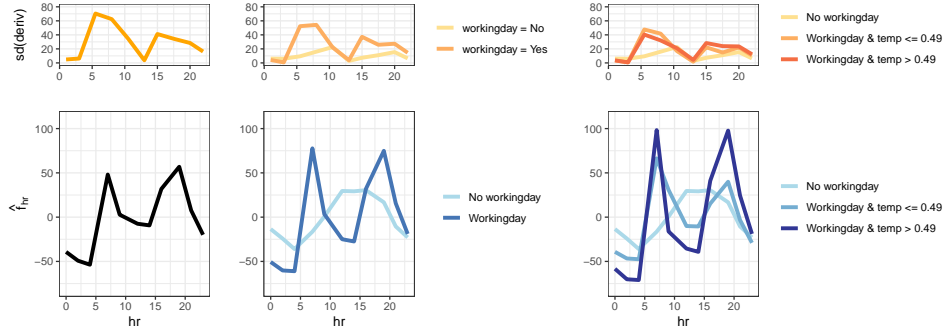


Figure 12: Global (left), regional ALE plots after applying GADGET for feature hour when $S = Z = \{hr, workingday\}$ (middle) and when $S = Z = \{hr, workingday, temp\}$ (right) of the bikesharing application. The upper plots show the interaction-related heterogeneity as defined in Section 4.4 and Appendix C.3.

8 Conclusion and Discussion

We introduced the new framework GADGET, which partitions the feature space into interpretable and distinct regions such that feature interactions of any subset of features are minimized. This allows to decompose joint effects into features’ main effects within the found regions. GADGET can be used with any feature effect method that satisfies the *local decomposability* axiom (Axiom 1). We showed applicability to the well-known global feature effect methods—namely PD, ALE, and SD—and provide respective estimates and visualizations for regional feature effects and interaction-related heterogeneity. Furthermore, we introduced different measures to quantify and analyze feature interactions by the reduction of interaction-related heterogeneity within GADGET. To define the interacting feature subset, we introduced PINT, a novel permutation-based significance test to detect global feature interactions that is applicable to any feature effect method used within GADGET.

Our experiments showed that PINT is often able to detect the true interacting feature subset in the presence of spurious interactions and that the pre-selection thus leads to more meaningful and interpretable results in GADGET. Moreover, considering feature effect methods within GADGET that are based on conditional distribution tend to lead to more stable results compared to considering feature effect methods based on marginal distribution when features are highly correlated. Furthermore, due to a different weighting scheme in the decomposition of Shapely values compared to the other considered feature effect methods, higher-order terms are less likely to be detected by GADGET, especially if Shapley values are not recalculated after each partitioning step. This approach might be computationally expensive, which can be seen as a possible limitation. However, recent research has focused on fast approximation techniques of Shapley values (e.g., Lundberg and Lee, 2017; Jethani et al., 2021; Lundberg et al., 2020; Chau et al., 2022) and may offer solutions to overcoming this limitation.

In general, our proposed method works well if learned feature interactions are not overly local and if observations can be grouped based on feature interactions such that local feature effects within the groups are homogeneous and, at the same time, show heterogeneous feature effects between different groups. With that, we can avoid an aggregation bias of global feature effect methods, obtain more insights into the learned effects, and may detect potential biases within different subgroups (as illustrated in Section 7). One of the real-world examples also showed that the frequently-made assumption of symmetric feature interactions (e.g., in Shapley values) is not always the case (see also Masoomi et al., 2023, for research on asymmetrical feature interactions). Thus, including domain knowledge to define the interacting feature subset Z might sometimes be meaningful.

Acknowledgments and Disclosure of Funding

This work has been partially supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy as part of the program “Bayerischen Verbundförderprogramms (BayVFP) – Förderlinie Digitalisierung – Förderbereich Informations- und Kommunikationstechnik” under the grant DIK-2106-0007 // DIK0260/02. The authors of this work take full responsibility for its content.

Appendix A. Details on M Plot

The M plot for feature \mathbf{x}_j at feature value x_j is defined as the marginal effect of feature \mathbf{x}_j using the conditional distribution (compared to the marginal distribution used in PD functions):

$$f_j^M(x_j) = \mathbb{E}[\hat{f}(X_j, X_{-j}) | X_j = x_j] = \int \hat{f}(x_j, \mathbf{x}_{-j}) d\mathbb{P}(\mathbf{x}_{-j} | x_j).$$

Hence, similar to PD plots, the local feature effect of M plots are also predictions at a specific feature value of \mathbf{x}_j but w.r.t. the conditional distribution $\mathbb{P}(\mathbf{x}_{-j} | x_j)$. Therefore, M plots can be seen as an average over ICE curves, which are restricted based on the given correlation structure. This leads to the inclusion of the feature effects of correlated features, as illustrated in the following example.

We draw 1000 samples of two multivariate normally distributed random variables X_1 and X_2 with $\mu_1 = \mu_2 = 0$, $\sigma_1 = \sigma_2 = 1$ and $\sigma_{12} = 0.9$. The true data-generating process is given by $y = -\mathbf{x}_1 + 2\mathbf{x}_2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.2)$. We train a linear model on the given data set. Based on the learned effects of the linear model, we would assume that the feature \mathbf{x}_1 has a negative influence on the predictions, as shown by the PD plot in Figure 13 (left). On the other hand, the M Plot accounts for the effect of the correlated feature \mathbf{x}_2 , which has a positive effect and (in absolute terms) a higher influence on the predictions than \mathbf{x}_1 , which leads to a positive slope in the right plot of Figure 13.

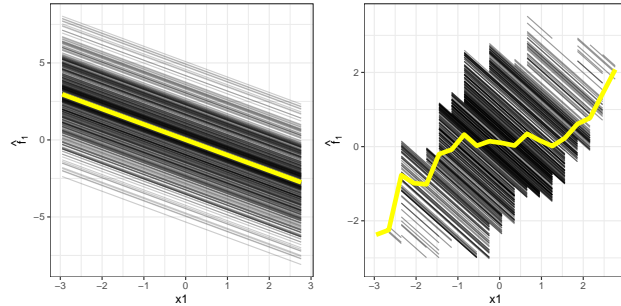


Figure 13: PD plot (left) and M Plot (right) for feature \mathbf{x}_1 .

Appendix B. Theoretical Evidence of GADGET

In this appendix, we provide the proofs for the theorems and the corollary of Section 4.2. Furthermore, we show the applicability of the feature effect methods PD, ALE, and SD within the GADGET algorithm by defining respective local feature effect functions that fulfill Axiom 1.

B.1 Proof of Theorem 2

Proof Sketch If the function $\hat{f}(\mathbf{x})$ can be decomposed as in Eq. (1) and if Axiom 1 holds for the local feature effect function h , then the main effect of feature \mathbf{x}_j at x_j is cancelled

DECOMPOSING GLOBAL FEATURE EFFECTS

out within the loss function in Eq. (10). Thus, the loss function measures the interaction-related heterogeneity of feature \mathbf{x}_j at x_j , since the variability of local effects in the subspace \mathcal{A}_g are only based on feature interactions between the j -th feature and features in $-j$.

Proof

$$\begin{aligned}
 \mathcal{L}(\mathcal{A}_g, x_j) &= \sum_{i: \mathbf{x}^{(i)} \in \mathcal{A}_g} \left(h(x_j, \mathbf{x}_{-j}^{(i)}) - \mathbb{E}[h(x_j, X_{-j}) | \mathcal{A}_g] \right)^2 \\
 &= \sum_{i: \mathbf{x}^{(i)} \in \mathcal{A}_g} \left(g_j(x_j) + \sum_{k=1}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=k}} g_{W \cup j}(x_j, \mathbf{x}_W^{(i)}) - \mathbb{E}[g_j(x_j) + \sum_{k=1}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=k}} g_{W \cup j}(x_j, X_W) | \mathcal{A}_g] \right)^2 \\
 &= \sum_{i: \mathbf{x}^{(i)} \in \mathcal{A}_g} \left(\sum_{k=1}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=k}} g_{W \cup j}(x_j, \mathbf{x}_W^{(i)}) - \mathbb{E}[g_{W \cup j}(x_j, X_W) | \mathcal{A}_g] \right)^2
 \end{aligned}$$

■

B.2 Proof of Theorem 3

Proof Sketch We show in two steps that the objective in Algorithm 1 minimizes interaction-related heterogeneity between features in S and Z . First, if the function $\hat{f}(\mathbf{x})$ can be decomposed as in Eq. (1) and if Axiom 1 holds for the local feature effect function h , then we can show (based on Theorem 2) that the objective $\mathcal{I}(t, z)$ in Algorithm 1 is defined by feature interactions between each feature $j \in S$ and features in $-j$. Second, since we only consider features in Z for splitting and thus minimize the interaction-related heterogeneity of features in S , we can show that feature interactions between features in S and features in $-Z$ are independent of the partitioning in Algorithm 1 (if all features contained in Z and all features in $-Z$ are pair-wise independent) and thus are not directly minimized in the objective $\mathcal{I}(t, z)$.

Proof

$$\begin{aligned}
 \mathcal{I}(t, z) &= \sum_{j \in S} \sum_{b \in \{l, r\}} (\mathcal{R}(\mathcal{A}_b^{t, z}, \mathbf{x}_j)) \\
 &\stackrel{\text{Eq. (11)}}{=} \sum_{j \in S} \sum_{b \in \{l, r\}} \left(\sum_{\substack{k: k \in \{1, \dots, m\} \\ \wedge x_j^{(k)} \in \mathcal{A}_l^{t, z}}} \mathcal{L}(\mathcal{A}_b^{t, z}, x_j^{(k)}) \right) \\
 &\stackrel{\text{Eq. (10)}}{=} \sum_{j \in S} \sum_{b \in \{l, r\}} \sum_{\substack{k: k \in \{1, \dots, m\} \\ \wedge x_j^{(k)} \in \mathcal{A}_b^{t, z}}} \sum_{i: \mathbf{x}^{(i)} \in \mathcal{A}_b^{t, z}} \left(h(x_j^{(k)}, \mathbf{x}_{-j}^{(i)}) - \mathbb{E}[h(x_j^{(k)}, X_{-j}) | \mathcal{A}_b^{t, z}] \right)^2 \\
 &\stackrel{\text{T. 2}}{=} \sum_{j \in S} \sum_{b \in \{l, r\}} \sum_{\substack{k: k \in \{1, \dots, m\} \\ \wedge x_j^{(k)} \in \mathcal{A}_b^{t, z}}} \sum_{i: \mathbf{x}^{(i)} \in \mathcal{A}_b^{t, z}} \left(\sum_{l=1}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=l}} g_{W \cup j}(x_j, \mathbf{x}_W^{(i)}) - \mathbb{E}[g_{W \cup j}(x_j, X_W) | \mathcal{A}_b^{t, z}] \right)^2
 \end{aligned}$$

The objective function $\mathcal{I}(t, z)$ is defined by the interaction-related heterogeneity between all features $j \in S \subseteq \{1, \dots, p\}$ and features in $-j$ for the sum of the left and right subspace $\mathcal{A}_l^{t,z}$ and $\mathcal{A}_r^{t,z}$. Since the aim is to minimize this term, we choose the split feature $z \in Z \subseteq \{1, \dots, p\}$ and split point t , which minimizes the sum of the risk values. However, since we only split w.r.t. features in Z and not w.r.t. feature in $-Z$, the objective focuses on minimizing the heterogeneity between features in S and features in Z , while interactions between features in S and features in $-Z$ are generally not minimized, since the heterogeneity is independent of the subspace $\mathcal{A}_b^{t,z}$.²¹ This can be shown by decomposing the risk function for features $j \in S$ into interaction effects between feature \mathbf{x}_j and features in Z (first term), interaction effects between feature \mathbf{x}_j and features in $-j$ with at least one feature of the subset Z and at least one feature of the subset $-Z$ (second term), and interaction effects between feature \mathbf{x}_j and features in $-Z$ (third term):

$$\begin{aligned} \mathcal{R}(\mathcal{A}_b^{t,z}, \mathbf{x}_j) &= \sum_{\substack{k: k \in \{1, \dots, m\} \\ \wedge \mathbf{x}_j^{(k)} \in \mathcal{A}_b^{t,z}}} \sum_{i: \mathbf{x}^{(i)} \in \mathcal{A}_b^{t,z}} \left(\sum_{l=1}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=l}} g_{W \cup j}(x_j, \mathbf{x}_W^{(i)}) - \mathbb{E}[g_{W \cup j}(x_j, X_W) | \mathcal{A}_b^{t,z}] \right)^2 \\ &= \sum_{\substack{k: k \in \{1, \dots, m\} \\ \wedge \mathbf{x}_j^{(k)} \in \mathcal{A}_b^{t,z}}} \sum_{i: \mathbf{x}^{(i)} \in \mathcal{A}_b^{t,z}} \left(\left(\sum_{l=1}^{|Z \setminus j|} \sum_{\substack{Z_l \subseteq Z \setminus j, \\ |Z_l|=l}} g_{Z_l \cup j}(x_j, \mathbf{x}_{Z_l}^{(i)}) - \mathbb{E}[g_{Z_l \cup j}(x_j, X_{Z_l}) | \mathcal{A}_b^{t,z}] \right) \right. \\ &+ \left(\sum_{l=2}^{p-1} \sum_{\substack{W \subseteq -j \\ \wedge \exists Z_l \subseteq Z \setminus j: Z_l \subset W \\ \wedge \exists -Z_l \subseteq -Z \setminus j: -Z_l \subset W, \\ |W|=l}} g_{W \cup j}(x_j, \mathbf{x}_{Z_l}^{(i)}, \mathbf{x}_{-Z_l}^{(i)}) - \mathbb{E}[g_{W \cup j}(x_j, X_{Z_l}, X_{-Z_l}) | \mathcal{A}_b^{t,z}] \right) \\ &+ \left. \left(\sum_{l=1}^{|-Z \setminus j|} \sum_{\substack{-Z_l \subseteq -Z \setminus j, \\ |-Z_l|=l}} g_{-Z_l \cup j}(x_j, \mathbf{x}_{-Z_l}^{(i)}) - \mathbb{E}[g_{-Z_l \cup j}(x_j, X_{-Z_l}) | \mathcal{A}_b^{t,z}] \right) \right)^2 \end{aligned}$$

While the first two terms contain feature interactions between j and features in Z , the last term does not. Furthermore, this term is (at least, in an uncorrelated setting) independent of the regarded subspace $\mathcal{A}_b^{t,z}$ and, thus, is not directly minimized by the given objective:

$$\left(\sum_{l=1}^{|-Z \setminus j|} \sum_{\substack{-Z_l \subseteq -Z \setminus j, \\ |-Z_l|=l}} g_{-Z_l \cup j}(x_j, \mathbf{x}_{-Z_l}^{(i)}) - \mathbb{E}[g_{-Z_l \cup j}(x_j, X_{-Z_l}) | \mathcal{A}_b^{t,z}] \right) \perp \mathcal{A}_b^{t,z}$$

since $\mathbb{E}[g_{-Z_l \cup j}(x_j, X_{-Z_l}) | \mathcal{A}_b^{t,z}] = \mathbb{E}[g_{-Z_l \cup j}(x_j, X_{-Z_l})]$. \blacksquare

B.3 Proof of Corollary 4

Proof Sketch Based on Theorem 3, we can show that the theoretical minimum of the objective is $\mathcal{I}(t^*, z^*) = 0$ if no feature in S interacts with any feature in $-Z$. In the following

21. This might be different if features in Z and $-Z$ are highly correlated.

DECOMPOSING GLOBAL FEATURE EFFECTS

proof, we apply the same decomposition of the risk function $\mathcal{R}(\mathcal{A}_b^{t,z}, \mathbf{x}_j)$, as done in the proof of Theorem 3. Since we assume that no feature in S interacts with any feature in $-Z$, the second and third term—which contain interactions between these two feature subsets—are zero. Hence, the risk function is only defined by feature interactions between features in S and Z , which are minimized by the objective in Algorithm 1.

Proof If the feature subset Z contains all features interacting with features in S , and hence no feature in $-Z$ interacts with any feature in S , then (w.r.t. the decomposition of the risk function in the proof of Theorem 3) the risk function for feature \mathbf{x}_j within a subspace $\mathcal{A}_b^{t,z}$ reduces to the variance of feature interactions between feature \mathbf{x}_j and features in Z :

$$\begin{aligned}
 \mathcal{R}(\mathcal{A}_b^{t,z}, \mathbf{x}_j) &= \sum_{\substack{k: k \in \{1, \dots, m\} \\ \wedge \mathbf{x}_j^{(k)} \in \mathcal{A}_b^{t,z}}} \sum_{i: \mathbf{x}^{(i)} \in \mathcal{A}_b^{t,z}} \left(\left(\sum_{l=1}^{|Z \setminus j|} \sum_{\substack{Z_l \subseteq Z \setminus j, \\ |Z_l|=l}} g_{Z_l \cup j}(x_j, \mathbf{x}_{Z_l}^{(i)}) - \mathbb{E}[g_{Z_l \cup j}(x_j, X_{Z_l}) | \mathcal{A}_b^{t,z}] \right) \right. \\
 &+ \left(\sum_{l=2}^{p-1} \sum_{\substack{W \subseteq -j \\ \wedge \exists Z_l \subseteq Z \setminus j; Z_l \subset W \\ \wedge \exists -Z_l \subseteq -Z \setminus j; -Z_l \subset W, \\ |W|=l}} \underbrace{g_{W \cup j}(x_j, \mathbf{x}_{Z_l}^{(i)}, \mathbf{x}_{-Z_l}^{(i)}) - \mathbb{E}[g_{W \cup j}(x_j, X_{Z_l}, X_{-Z_l}) | \mathcal{A}_b^{t,z}]}_{=0} \right) \\
 &+ \left. \left(\sum_{l=1}^{|-Z \setminus j|} \sum_{\substack{-Z_l \subseteq -Z \setminus j, \\ |-Z_l|=l}} \underbrace{g_{-Z_l \cup j}(x_j, \mathbf{x}_{-Z_l}^{(i)}) - \mathbb{E}[g_{-Z_l \cup j}(x_j, X_{-Z_l}) | \mathcal{A}_b^{t,z}]}_{=0} \right) \right)^2 \\
 &= \sum_{\substack{k: k \in \{1, \dots, m\} \\ \wedge \mathbf{x}_j^{(k)} \in \mathcal{A}_b^{t,z}}} \sum_{i: \mathbf{x}^{(i)} \in \mathcal{A}_b^{t,z}} \left(\sum_{l=1}^{|Z \setminus j|} \sum_{\substack{Z_l \subseteq Z \setminus j, \\ |Z_l|=l}} g_{Z_l \cup j}(x_j, \mathbf{x}_{Z_l}^{(i)}) - \mathbb{E}[g_{Z_l \cup j}(x_j, X_{Z_l}) | \mathcal{A}_b^{t,z}] \right)^2
 \end{aligned}$$

Since the objective is defined such that it minimizes these interactions for all $j \in S$ by splitting the feature space w.r.t. features in Z , we can split deep enough to achieve $g_{Z_l \cup j}(x_j, \mathbf{x}_{Z_l}^{(i)}) = \mathbb{E}[g_{Z_l \cup j}(x_j, X_{Z_l}) | \mathcal{A}_b^{t,z}]$ for all terms within the sums of the risk function and for all $j \in S$. In other words, the individual interaction effect is equal to the expected interaction effect within a subspace, and thus the theoretical minimum of the objective is $\mathcal{I}(t^*, z^*) = 0$. ■

B.4 Applicability of PD within GADGET

Here, we show how h must be defined to fulfill Axiom 1 defined in Section 4.2 for the feature effect method PD.

Local Decomposition: The local feature effect method used in PDs are ICE curves. The i -th ICE curve of feature \mathbf{x}_j can be decomposed as follows:

$$\begin{aligned}
 \hat{f}(\mathbf{x}_j, \mathbf{x}_{-j}^{(i)}) &= \underbrace{g_0}_{\text{constant term}} + \underbrace{g_j(\mathbf{x}_j)}_{\text{main effect of } \mathbf{x}_j} + \underbrace{\sum_{k \in -j} g_k(\mathbf{x}_k^{(i)})}_{\substack{\text{main effect of all other} \\ \text{features in } -j \text{ for observation } i}} \\
 &+ \underbrace{\sum_{k=1}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=k}} g_{W \cup \{j\}}(\mathbf{x}_j, \mathbf{x}_W^{(i)})}_{\substack{(k+1)\text{-order interaction between} \\ \mathbf{x}_j \text{ and features in } -j \text{ for observation } i}} + \underbrace{\sum_{k=2}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=k}} g_W(\mathbf{x}_W^{(i)})}_{\substack{k\text{-order interaction between} \\ \text{features in } -j \text{ for observation } i}}
 \end{aligned}$$

However, this decomposition of the local feature effect of \mathbf{x}_j contains not only feature effects that depend on \mathbf{x}_j , but also other effects (e.g., i -th main effects of features in $-j$), and thus Axiom 1 is not fulfilled by ICE curves. However, by mean-centering ICE curves, constant and feature effects independent of \mathbf{x}_j are cancelled out, and thus $\hat{f}^c(\mathbf{x}_j, \mathbf{x}_{-j}^{(i)})$ can be decomposed into the mean-centered main effect of \mathbf{x}_j and the i -th mean-centered interaction effect between \mathbf{x}_j and features in $-j$. Hence, the mean-centered ICE of feature \mathbf{x}_j at x_j can be decomposed as follows:

$$\begin{aligned}
 \hat{f}^c(x_j, \mathbf{x}_{-j}^{(i)}) &= \hat{f}(x_j, \mathbf{x}_{-j}^{(i)}) - \mathbb{E}[\hat{f}(X_j, \mathbf{x}_{-j}^{(i)})] \\
 &= g_0 + g_j(x_j) + \sum_{k \in -j} g_k(\mathbf{x}_k^{(i)}) + \sum_{k=1}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=k}} g_{W \cup \{j\}}(x_j, \mathbf{x}_W^{(i)}) + \sum_{k=2}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=k}} g_W(\mathbf{x}_W^{(i)}) \\
 &- g_0 - E[g_j(X_j)] - \sum_{k \in -j} g_k(\mathbf{x}_k^{(i)}) - E \left[\sum_{k=1}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=k}} g_{W \cup \{j\}}(X_j, \mathbf{x}_W^{(i)}) \right] - \sum_{k=2}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=k}} g_W(\mathbf{x}_W^{(i)}) \\
 &= g_j(x_j) - E[g_j(X_j)] + \sum_{k=1}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=k}} g_{W \cup \{j\}}(x_j, \mathbf{x}_W^{(i)}) - E \left[\sum_{k=1}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=k}} g_{W \cup \{j\}}(X_j, \mathbf{x}_W^{(i)}) \right] \\
 &= \underbrace{g_j^c(x_j)}_{\substack{\text{mean-centered} \\ \text{main effect of } \mathbf{x}_j}} + \underbrace{\sum_{k=1}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=k}} g_{W \cup \{j\}}^c(x_j, \mathbf{x}_W^{(i)})}_{\substack{\text{mean-centered interaction effect of} \\ \mathbf{x}_j \text{ with } \mathbf{x}_{-j}^{(i)}}}
 \end{aligned}$$

Thus, Axiom 1 is satisfied by mean-centered ICE curves and can be used as local feature effect h within GADGET. Following from that, the mean-centered PD for feature \mathbf{x}_j at x_j can be decomposed by:

$$f_j^{PD,c}(x_j) = \mathbb{E}[\hat{f}^c(x_j, X_{-j})] = g_j^c(x_j) + \sum_{k=1}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=k}} E \left[g_{W \cup \{j\}}^c(x_j, X_W) \right],$$

which is the mean-centered main effect of feature \mathbf{x}_j and the expected mean-centered interaction effect with feature \mathbf{x}_j at feature value x_j . Based on these decompositions and for

DECOMPOSING GLOBAL FEATURE EFFECTS

$h = \hat{f}^c(x_j, \mathbf{x}_{-j}^{(i)})$, we can show that the loss function only depends on the feature interaction effect between the j -th feature and features in $-j$ (Theorem 2):

$$\begin{aligned} \mathcal{L}^{PD}(\mathcal{A}_g, x_j) &= \sum_{i: \mathbf{x}^{(i)} \in \mathcal{A}_g} \left(\hat{f}^c(x_j, \mathbf{x}_{-j}^{(i)}) - \mathbb{E}[\hat{f}^c(x_j, X_{-j}) | \mathcal{A}_g] \right)^2 \\ &= \sum_{i: \mathbf{x}^{(i)} \in \mathcal{A}_g} \left(g_j^c(x_j) + \sum_{k=1}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=k}} g_{W \cup \{j\}}^c(x_j, \mathbf{x}_W^{(i)}) \right. \\ &\quad \left. - g_j^c(x_j) - \sum_{k=1}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=k}} \mathbb{E}[g_{W \cup \{j\}}^c(x_j, X_W) | \mathcal{A}_g] \right)^2 \\ &= \sum_{i: \mathbf{x}^{(i)} \in \mathcal{A}_g} \left(\sum_{k=1}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=k}} g_{W \cup \{j\}}^c(x_j, \mathbf{x}_W^{(i)}) - \mathbb{E}[g_{W \cup \{j\}}^c(x_j, X_W) | \mathcal{A}_g] \right)^2 \end{aligned}$$

REPID as special case of GADGET. The objective function $\mathcal{I}(t, z)$ in Algorithm 1 for $h = \hat{f}^c(x_j, \mathbf{x}_{-j}^{(i)})$ is defined by the above loss function $\mathcal{L}^{PD}(\mathcal{A}_g, x_j)$ as follows:

$$\mathcal{I}(t, z) = \sum_{j \in S} \sum_{g \in \{l, r\}} \sum_{k: k \in \{1, \dots, m\} \wedge \mathbf{x}_j^{(k)} \in \mathcal{A}_g} \mathcal{L}^{PD}(\mathcal{A}_g, \mathbf{x}_j^{(k)})$$

For the special case where we consider one feature of interest that we want to visualize ($S = j$) and all other features as possible split features ($Z = -j$), the objective function of GADGET reduces to:

$$\mathcal{I}(t, z) = \sum_{g \in \{l, r\}} \sum_{k=1}^m \mathcal{L}^{PD}(\mathcal{A}_g, \mathbf{x}_j^{(k)}),$$

which is the same objective used within REPID. Thus, for the special case where we choose mean-centered ICE curves as local feature effect method and $S = j$ and $Z = -j$, GADGET is equivalent to REPID.

B.5 Applicability of ALE Within GADGET

Here, we show the fulfillment of Axiom 1 defined in Section 4.2 for the underlying local feature effect function in ALE.

Local Decomposition: The local feature effect method used in ALE is the partial derivative of the prediction function at $\mathbf{x}_j = x_j$. Thus, we define the local feature effect function h by $h(x_j, \mathbf{x}_{-j}^{(i)}) := \frac{\partial \hat{f}(x_j, \mathbf{x}_{-j}^{(i)})}{\partial x_j}$. We can decompose h such that it only depends on main and interaction effects of and with feature \mathbf{x}_j :

$$\begin{aligned} \frac{\partial \hat{f}(x_j, \mathbf{x}_{-j}^{(i)})}{\partial x_j} &= \frac{\partial \left(g_0 + \sum_{j=1}^p g_j(x_j) + \sum_{j \neq k} g_{jk}(x_j, \mathbf{x}_k^{(i)}) + \dots + g_{12\dots p}(\mathbf{x}^{(i)}) \right)}{\partial x_j} \\ &= \frac{\partial g_j(x_j)}{\partial x_j} + \sum_{k=1}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=k}} \frac{\partial g_{W \cup j}(\mathbf{x}_j, \mathbf{x}_W^{(i)})}{\partial x_j} \end{aligned}$$

Taking the conditional expectation over the local feature effects (partial derivatives) at x_j yields the (conditional) expected (i.e., global) feature effect at x_j :

$$\mathbb{E} \left[\frac{\partial \hat{f}(X_j, X_{-j})}{\partial x_j} \middle| X_j = x_j \right] = \frac{\partial g_j(x_j)}{\partial x_j} + \sum_{k=1}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=k}} \mathbb{E} \left[\frac{\partial g_{W \cup j}(X_j, X_W)}{\partial x_j} \middle| X_j = x_j \right]$$

Based on these decompositions, we can show that the loss function for ALE only depends on the feature interaction effect between the j -th feature and features in $-j$ (Theorem 2):

$$\begin{aligned} \mathcal{L}^{ALE}(\mathcal{A}_g, x_j) &= \sum_{\substack{i: \mathbf{x}^{(i)} \in \mathcal{A}_g \wedge \\ \mathbf{x}^{(i)} \in \mathcal{P}(\mathbf{x}_{-j} | x_j)}} \left(\frac{\partial \hat{f}(x_j, \mathbf{x}_{-j}^{(i)})}{\partial x_j} - \mathbb{E} \left[\frac{\partial \hat{f}(X_j, X_{-j})}{\partial x_j} \middle| \mathcal{A}_g \wedge X_j = x_j \right] \right)^2 \\ &= \sum_{\substack{i: \mathbf{x}^{(i)} \in \mathcal{A}_g \wedge \\ \mathbf{x}^{(i)} \in \mathcal{P}(\mathbf{x}_{-j} | x_j)}} \left(\frac{\partial g_j(x_j)}{\partial x_j} + \sum_{k=1}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=k}} \frac{\partial g_{W \cup j}(\mathbf{x}_j, \mathbf{x}_W^{(i)})}{\partial x_j} \right. \\ &\quad \left. - \frac{\partial g_j(x_j)}{\partial x_j} - \sum_{k=1}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=k}} \mathbb{E} \left[\frac{\partial g_{W \cup j}(X_j, X_W)}{\partial x_j} \middle| \mathcal{A}_g \wedge X_j = x_j \right] \right)^2 \\ &= \sum_{\substack{i: \mathbf{x}^{(i)} \in \mathcal{A}_g \wedge \\ \mathbf{x}^{(i)} \in \mathcal{P}(\mathbf{x}_{-j} | x_j)}} \left(\sum_{k=1}^{p-1} \sum_{\substack{W \subseteq -j, \\ |W|=k}} \left(\frac{\partial g_{W \cup j}(\mathbf{x}_j, \mathbf{x}_W^{(i)})}{\partial x_j} - \mathbb{E} \left[\frac{\partial g_{W \cup j}(X_j, X_W)}{\partial x_j} \middle| \mathcal{A}_g \wedge X_j = x_j \right] \right) \right)^2 \end{aligned}$$

B.6 Applicability of SD Within GADGET

Here, we show the fulfillment of Axiom 1 defined in Section 4.2 for Shapley values, which are the underlying local feature effect in SD plots.

Local Decomposition: The local feature effect function in the SD plot is the Shapley value. We define $h(x_j, \mathbf{x}_{-j}^{(i)}) := \phi_j^{(i)}(x_j)$ to be the Shapley value for the i -th local feature effect at a fixed value x_j , which is typically the i -th feature value of \mathbf{x}_j (i.e., $x_j = \mathbf{x}_j^{(i)}$). In Eq. (8),

DECOMPOSING GLOBAL FEATURE EFFECTS

we defined $\phi_j^{(i)}(x_j)$ according to Herren and Hahn (2022) by the following decomposition:

$$\begin{aligned}\phi_j^{(i)}(x_j) &= \sum_{k=0}^{p-1} \frac{1}{k+1} \sum_{\substack{W \subseteq -j: \\ |W|=k}} \left(\mathbb{E}[\hat{f}(x_j, X_{-j}) | X_W = \mathbf{x}_W^{(i)}] - \sum_{V \subset \{W \cup j\}} \mathbb{E}[\hat{f}(X) | X_V = \mathbf{x}_V^{(i)}] \right) \\ &= g_j^c(x_j) + \sum_{k=1}^{p-1} \frac{1}{k+1} \sum_{W \subseteq -j: |W|=k} g_{W \cup j}^c(x_j, \mathbf{x}_W^{(i)}),\end{aligned}$$

with $g_{W \cup j}^c(x_j, \mathbf{x}_W^{(i)}) = \mathbb{E}[\hat{f}(x_j, X_{-j}) | X_W = \mathbf{x}_W^{(i)}] - \sum_{V \subset \{W \cup j\}} \mathbb{E}[\hat{f}(X) | X_V = \mathbf{x}_V^{(i)}]$. Hence, we can decompose h such that it only depends on main effects of and interaction effects with feature \mathbf{x}_j .

Taking the expectation over the local feature effects $h = \phi_j$ at x_j yields the expected (i.e., global) feature effect of Shapley values at $\mathbf{x}_j = x_j$.

$$\mathbb{E}_{X_W}[\phi_j(x_j)] = g_j^c(x_j) + \sum_{k=1}^{p-1} \frac{1}{k+1} \sum_{W \subseteq -j: |W|=k} \mathbb{E}[g_{W \cup j}^c(x_j, X_W)]$$

Based on these decompositions, we can show that the loss function for SD only depends on feature interactions between the j -th feature and features in $-j$ (Theorem 2):

$$\begin{aligned}\mathcal{L}^{SD}(\mathcal{A}_g, x_j) &= \sum_{i: \mathbf{x}^{(i)} \in \mathcal{A}_g \wedge \mathbf{x}_j^{(i)} = x_j} \left(\phi_j^{(i)}(x_j) - \mathbb{E}_{X_W}[\phi_j(x_j) | \mathcal{A}_g] \right)^2 \\ &= \sum_{i: \mathbf{x}^{(i)} \in \mathcal{A}_g \wedge \mathbf{x}_j^{(i)} = x_j} \left(g_j^c(x_j) + \sum_{k=1}^{p-1} \frac{1}{k+1} \sum_{W \subseteq -j: |W|=k} g_{W \cup j}^c(x_j, \mathbf{x}_W^{(i)}) \right. \\ &\quad \left. - g_j^c(x_j) + \sum_{k=1}^{p-1} \frac{1}{k+1} \sum_{W \subseteq -j: |W|=k} \mathbb{E}[g_{W \cup j}^c(x_j, X_W) | \mathcal{A}_g] \right)^2 \\ &= \sum_{i: \mathbf{x}^{(i)} \in \mathcal{A}_g \wedge \mathbf{x}_j^{(i)} = x_j} \left(\sum_{k=1}^{p-1} \frac{1}{k+1} \sum_{W \subseteq -j: |W|=k} \left(g_{W \cup j}^c(x_j, \mathbf{x}_W^{(i)}) - \mathbb{E}[g_{W \cup j}^c(x_j, X_W) | \mathcal{A}_g] \right) \right)^2\end{aligned}$$

Appendix C. Further Characteristics of Feature Effect Methods

In this section, we cover further characteristics of the different feature effect methods used within GADGET. As illustrated for PD in Section 4.3, we also show here for ALE and SD that the joint feature effect (and possibly the prediction function) within the final regions of GADGET can be approximated by the sum of univariate feature effects. Hence, the joint feature effect can be additively decomposed into the features' main effects within the final regions. Furthermore, we provide an overview on estimates and visualization techniques for the regional feature effects and interaction-related heterogeneity for the different feature effect methods. We also explain how categorical features are handled within GADGET, depending on the underlying feature effect method.

C.1 Decomposability of ALE

Figure 14 shows the effect plots when GADGET is applied to the uncorrelated simulation example of Section 3 when ALE (the underlying derivatives) and $S = Z = \{1, 2, 3\}$ are used. The grey curves before the split illustrate the global ALE curves. Since they do not show the interaction-related heterogeneity of the underlying local effects, we added a plot that visualizes this heterogeneity by providing the standard deviation of the derivatives within each interval as a (yellow) curve along the range of \mathbf{x}_j , similar to the idea of Goldstein et al. (2015) for derivative ICE curves. This shows us that local effects for feature \mathbf{x}_2 are very homogeneous over the entire range of \mathbf{x}_2 , while the local feature effects of \mathbf{x}_1 show a constant heterogeneous behavior and a regional high heterogeneity around $\mathbf{x}_3 = 0$ is visible for \mathbf{x}_3 . GADGET chooses $\mathbf{x}_3 = -0.003$ as the best split point that reduces the interaction-related heterogeneity of the three features almost completely. Thus, the ALE curves we receive in the subspaces are more representative for the underlying individuals.

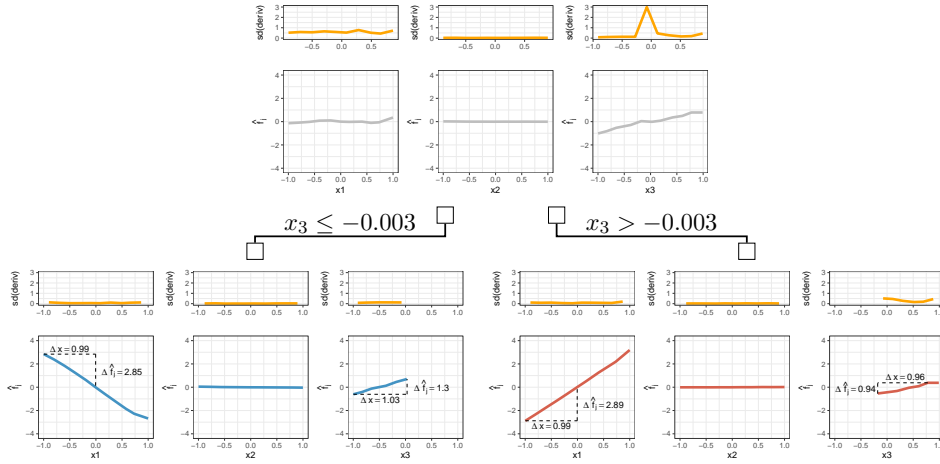


Figure 14: Visualization of applying GADGET with $S = Z = \{1, 2, 3\}$ to derivatives of ALE for the uncorrelated simulation example of Section 3 with $Y = 3X_1\mathbb{1}_{X_3>0} - 3X_1\mathbb{1}_{X_3\leq 0} + X_3 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.09)$. The upper plots show the standard deviation of the derivatives (yellow) and the ALE curves (grey) on the entire feature space, while the lower plots represent the respective standard deviation of the derivatives and regional ALE curves after partitioning the feature space w.r.t. $\mathbf{x}_3 = -0.003$.

Equivalently to PD plots, ALE plots also contain an additive recovery and, thus, can be decomposed additively in main and interaction effects (see Section 2.3). Furthermore, if Z is defined such that all features interacting with features in S are included and if GADGET is applied such that the theoretical minimum of the objective function is reached, then

DECOMPOSING GLOBAL FEATURE EFFECTS

according to Corollary 4, the joint mean-centered ALE function $f_{S|\mathcal{A}_g}^{ALE,c}$ within each final subspace \mathcal{A}_g can be decomposed into the 1-dimensional mean-centered ALE functions of features in S . Hence, since there are no more interactions between features in S and other features present in the final regions, $f_{S|\mathcal{A}_g}^{ALE,c}$ can be uniquely decomposed into the mean-centered main effects of features in S —just as in PD functions (Apley and Zhu, 2020):

$$f_{S|\mathcal{A}_g}^{ALE,c}(\mathbf{x}_S) = \sum_{j \in S} f_{j|\mathcal{A}_g}^{ALE,c}(\mathbf{x}_j) \quad (18)$$

Moreover, let $-S$ be the subset of features that do not interact with any other features. Then, according to Eq. (18) and Eq. (4), the prediction function $\hat{f}_{\mathcal{A}_g}$ within the region \mathcal{A}_g can be decomposed into the 1-dimensional mean-centered ALE functions of all p features, plus some constant value g_0 :

$$\hat{f}_{\mathcal{A}_g}(\mathbf{x}) = g_0 + \sum_{j=1}^p f_{j|\mathcal{A}_g}^{ALE,c}(\mathbf{x}_j).$$

We can again derive this decomposition from Figure 14, where \mathbf{x}_2 shows an effect of 0 with low heterogeneity before and after the split. The feature effects of \mathbf{x}_1 and \mathbf{x}_3 show high heterogeneity before the split, which is almost completely minimized after the split w.r.t. \mathbf{x}_3 . Hence, the resulting regional (linear) ALE curves are representative estimates for the underlying local effects. Therefore, we can approximate the prediction function within each subspace by

$$\hat{f}_{\mathcal{A}_l}(\mathbf{x}) = g_0 + \frac{-2.85}{0.99}\mathbf{x}_1 + \frac{1.3}{1.03}\mathbf{x}_3 = g_0 - 2.89\mathbf{x}_1 + 1.26\mathbf{x}_3$$

and

$$\hat{f}_{\mathcal{A}_r}(\mathbf{x}) = g_0 + \frac{2.89}{0.99}\mathbf{x}_1 + \frac{0.94}{0.96}\mathbf{x}_3 = g_0 + 2.92\mathbf{x}_1 + 0.98\mathbf{x}_3.$$

Particularities of ALE Estimation. As seen for the continuous feature \mathbf{x}_3 in the simulation example presented here, abrupt interactions (“jumps”)²² might be difficult to estimate for models that learn smooth effects, such as NNs (used here) or SVMs—especially when compared to models such as decision trees. Hence, depending on the model, these type of feature interactions can lead to very high partial derivatives in a region around the “jump” point instead of a high partial derivative at exactly the one specific “jump” point (here: $\mathbf{x}_3 = 0$), thus leading to non-reducible heterogeneity. This is illustrated in the upper right plot of Figure 14. The standard deviation of the derivatives of \mathbf{x}_3 are very high in the region around and not exactly at $\mathbf{x}_3 = 0$. This interaction-related heterogeneity should be (almost) completely reduced when splitting w.r.t. $\mathbf{x}_3 = 0$. However, high values will remain, since the model did not perfectly capture this kind of interaction. To account for this issue in the estimation and partitioning process within GADGET, we use the following procedure for continuous features: In the two new subspaces after a split, if the derivatives of feature values close to the split point vary at least twice as much (measured by the

22. With abrupt interaction, we mean interactions that lead to an abrupt change of the influence of one feature (\mathbf{x}_1) based on the influence of another feature at a specific (“jump”) point ($\mathbf{x}_3 = 0$) like the feature interaction between \mathbf{x}_1 and \mathbf{x}_3 in the here presented simulation example: $Y = 3X_1 \mathbb{1}_{X_3 > 0} - 3X_1 \mathbb{1}_{X_3 \leq 0} + X_3 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.09)$.

standard deviation) as the derivatives of the other observations within each subspace, then the derivatives of feature values close to the split point are replaced by values drawn from a normally distributed random variable where mean and variance are estimated by the derivatives of the remaining observation within each subspace.

C.2 Decomposability of SD

Recalculation Versus No Recalculation of Shapley Values. In Section 4.5, we argued that Shapley values must be recalculated after each partitioning step in order for each new subspace to receive SD effects in the final subspaces that are representative of the underlying main effects within each subspace. Meanwhile, the unconditional expected value (i.e., the feature interactions on the entire feature space) are minimized without recalculating the conditional expected values.

The difference in the final feature effects within the subspaces is illustrated when comparing the left plot of Figure 5 (split without recalculation) with the respective plots of feature \mathbf{x}_1 of Figure 15. Without recalculation, the effect of feature \mathbf{x}_1 is still regarded as an interaction effect between \mathbf{x}_1 and \mathbf{x}_3 , and hence only half of the joint interaction effect is assigned to \mathbf{x}_1 (i.e., the respective slope within the regions is 1.5 and -1.5 instead of 3 and -3), and the other half of the joint interaction effect is assigned to \mathbf{x}_3 . When Shapley values are recalculated after the first partitioning step within each subspace, we can see in Figure 15 that no more interactions are present between \mathbf{x}_1 and \mathbf{x}_3 within each subspace, due to the split w.r.t. \mathbf{x}_3 . Hence, the effect of \mathbf{x}_1 is recognized as the main effect with the slope approximately defined in the data-generating process. Furthermore, due to interactions with \mathbf{x}_1 , the heterogeneity of feature effects of \mathbf{x}_3 is also reduced after the split, owing to recalculation.

Note: If the feature we use for partitioning the feature space ($z \in Z$) coincides with the features of interest (S), then the Shapley values should be recalculated in Algorithm 1 to find the best split point (at least, if we choose the approach with recalculation after each partitioning step). The reason is that if $z \in S$, we also want to reduce the interaction-related heterogeneity within z that is not accounted for if we do not recalculate the Shapley values within the new subspace. For example, in Figure 15, we split according to \mathbf{x}_3 , which is also a feature of interest ($3 \in S$). If we do not recalculate the Shapley values for \mathbf{x}_3 within the splitting process, then the sum of the risk of any two subspaces for \mathbf{x}_3 will always be approximately the same as the risk of the parent node, and thus the heterogeneity reduction for \mathbf{x}_3 (which is shown in the regional plots of Figure 15) is not considered in the objective of Algorithm 1.

Decomposition. Herren and Hahn (2022) show that Shapley values can be decomposed by weighted PD functions (see Eq. 8)). Hence, if the global SD feature effect as defined in Eq. (16) is considered, the same decomposition rules as defined for PD plots apply. In other words, if Z contains all features that interact with features in S and if GADGET is applied such that the theoretical minimum of the objective function is reached, then according to Corollary 4, the following decomposition in 1-dimensional global SD effect functions of features in S holds:

$$f_{S|A_g}^{SD}(\mathbf{x}_S) = \sum_{j \in S} f_{j|A_g}^{SD}(\mathbf{x}_j). \quad (19)$$

DECOMPOSING GLOBAL FEATURE EFFECTS

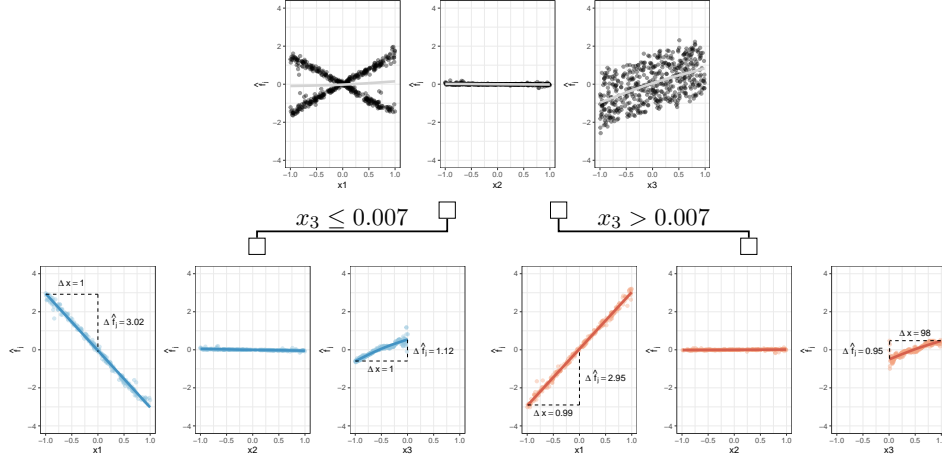


Figure 15: Visualization of applying GADGET with $S = Z = \{1, 2, 3\}$ to Shapley values of the uncorrelated simulation example of Section 3 with $Y = 3X_1 \mathbb{1}_{X_3 > 0} - 3X_1 \mathbb{1}_{X_3 \leq 0} + X_3 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.09)$. The upper plots show the Shapley values and the global estimated SD curve on the entire feature space, while the lower plots represent the respective Shapley values and regional SD curves after partitioning the feature space w.r.t. $\mathbf{x}_3 = 0.007$.

If all features containing heterogeneous effects (feature interactions) are included in the subset S , and the subset Z consists of all features that interact with features in S , then according to Eq. (19) and Eq. (4), the prediction function $\hat{f}_{\mathcal{A}_g}$ within the region \mathcal{A}_g can be uniquely decomposed into the 1-dimensional global SD effect functions of all p features, plus some constant value g_0 :

$$\hat{f}_{\mathcal{A}_g}(\mathbf{x}) = g_0 + \sum_{j=1}^p f_{j|\mathcal{A}_g}^{SD}(\mathbf{x}_j).$$

Again, we can derive this decomposition from Figure 14 in the same way we did for PD and ALE plots. Hence, we can approximate the prediction function within each subspace by $\hat{f}_{\mathcal{A}_l}(\mathbf{x}) = g_0 - 3.02\mathbf{x}_1 + 1.12\mathbf{x}_3$ and $\hat{f}_{\mathcal{A}_r}(\mathbf{x}) = g_0 + 2.98\mathbf{x}_1 + 1.03\mathbf{x}_3$.

Equivalence of SD and Mean-Centered PD. According to Herren and Hahn (2022), the Shapley value $\phi_j^{(i)}(x_j)$ of the i -th observation at $\mathbf{x}_j = x_j$ can be decomposed as defined in

Eq. (8) to

$$\begin{aligned}\phi_j^{(i)}(x_j) &= \sum_{k=0}^{p-1} \frac{1}{k+1} \sum_{\substack{W \subseteq -j: \\ |W|=k}} \left(\mathbb{E}[\hat{f}(x_j, X_{-j}) | X_W = \mathbf{x}_W^{(i)}] - \sum_{V \subset \{W \cup j\}} \mathbb{E}[\hat{f}(X) | X_V = \mathbf{x}_V^{(i)}] \right) \\ &= g_j^c(x_j) + \sum_{k=1}^{p-1} \frac{1}{k+1} \sum_{W \subseteq -j: |W|=k} g_{W \cup j}^c(x_j, \mathbf{x}_W^{(i)}),\end{aligned}$$

with $g_{W \cup j}^c(x_j, \mathbf{x}_W^{(i)}) = \mathbb{E}[\hat{f}(x_j, X_{-j}) | X_W = \mathbf{x}_W^{(i)}] - \sum_{V \subset \{W \cup j\}} \mathbb{E}[\hat{f}(X) | X_V = \mathbf{x}_V^{(i)}]$.

As in Eq. (16), the global feature effect (SD) of feature \mathbf{x}_j at x_j is then defined by

$$f_j^{SD}(x_j) = \mathbb{E}_{X_W}[\phi_j] = g_j^c(x_j) + \sum_{k=1}^{p-1} \frac{1}{k+1} \sum_{W \subseteq -j: |W|=k} \mathbb{E}[g_{W \cup j}^c(x_j, X_W)]$$

Hence, if Corollary 4 is satisfied, if the joint global SD effect of features in S can be decomposed into the univariate SD effects as in Eq. (19), and if the interventional approach for Shapley calculation is used, then all feature interactions are zero, and the global SD effect of feature \mathbf{x}_j at x_j is given by

$$\begin{aligned}f_j^{Shap}(x_j) &= g_j^c(x_j) + \sum_{k=1}^{p-1} \frac{1}{k+1} \sum_{W \subseteq -j: |W|=k} \mathbb{E}[g_{W \cup j}^c(x_j, X_W)] \\ &\stackrel{\text{Cor. 4}}{=} g_j^c(x_j) \\ &= \mathbb{E}[\hat{f}(x_j, X_{-j})] - \mathbb{E}[\hat{f}(X)],\end{aligned}$$

which is equivalent to the mean-centered PD of feature \mathbf{x}_j at x_j .

C.3 Overview on Estimates and Visualizations

We provide here an overview on the estimates and visualization techniques for PD, ALE, and SD within GADGET that we introduced in Sections 4.3-4.5.

Local Effect. The local effect h for a feature \mathbf{x}_j at feature value x_j used within GADGET is estimated by

- **PD:** mean-centered ICE $\hat{h}^{(i)} = \hat{f}^c(x_j, \mathbf{x}_{-j}^{(i)})$
- **ALE:** partial derivatives estimated by prediction differences within k -th interval $\hat{h}^{(i)} = \hat{f}(z_{k-1,j}, \mathbf{x}_{-j}^{(i)}) - \hat{f}(z_{k-1,j}, \mathbf{x}_{-j}^{(i)})$ where $x_j \in]z_{k-1,j}, z_{k,j}]$
- **SD:** Shapley value $\hat{h}^{(i)} = \hat{\phi}_j^{(i)}$

DECOMPOSING GLOBAL FEATURE EFFECTS

Regional Effect. The feature effect for a feature \mathbf{x}_j at feature value x_j within a subspace/region \mathcal{A}_g of GADGET is estimated by

- **PD:** mean-centered regional PD $\hat{f}_{j|\mathcal{A}_g}^{PD,c}(x_j) = \frac{1}{|N_g|} \sum_{i \in N_g} \hat{f}^c(x_j, \mathbf{x}_{-j}^{(i)})$ with N_g being the index set of all $i : \mathbf{x}^{(i)} \in \mathcal{A}_g$.
- **ALE:** regional ALE $\hat{f}_{j|\mathcal{A}_g}^{ALE}(x_j) = \sum_{k=1}^{k_j(x_j)} \frac{1}{|N_g(k)|} \sum_{i \in N_g(k)} \left[\hat{f}(z_{k,j}, \mathbf{x}_{-j}^{(i)}) - \hat{f}(z_{k-1,j}, \mathbf{x}_{-j}^{(i)}) \right]$ with $N_g(k)$ being the index set of all $i : x_j \in]z_{k-1,j}, z_{k,j}] \wedge \mathbf{x}^{(i)} \in \mathcal{A}_g$.
- **SD:** regional SD $\hat{f}_{j|\mathcal{A}_g}^{SD}(x_j)$ is estimated by fitting a GAM on $\{\mathbf{x}_j^{(i)}, \hat{\phi}_j^{(i)}\}_{i:\mathbf{x}^{(i)} \in \mathcal{A}_g}$.

The regional effect for feature \mathbf{x}_j is visualized for all $x_j \in \mathcal{A}_g$. Therefore, the respective GAM curve is plotted in the case of SD, while we linearly interpolate between the grid-wise/interval-wise estimates of PD/ALE to receive the regional effect curves.

Interaction-Related Heterogeneity. The interaction-related heterogeneity for a feature \mathbf{x}_j at feature value x_j within a subspace/region \mathcal{A}_g of GADGET is estimated by the loss function in Eq. (10), which quantifies the variance of local effects at x_j and is visualized by

- **PD:** 95% interval around (mean-centered) regional PD estimate $\left[\hat{f}_{j|\mathcal{A}_g}^{PD,c}(x_j) \pm 1.96 \cdot \sqrt{\hat{\mathcal{L}}^{PD}(\mathcal{A}_g, x_j)} \right]$.
- **ALE:** standard deviation of local effects $\sqrt{\hat{\mathcal{L}}^{ALE}(\mathcal{A}_g, x_j)}$.
- **SD:** Shapley values are recalculated within each region \mathcal{A}_g and plotted with the fitted GAM for the regional SD effect to visualize the variation of local feature effects aka interaction-related heterogeneity.

For each feature $j \in S$, we generate one figure showing the regional effect curves of all final regions we obtain after applying GADGET. For PD, the regional effect curves are accompanied with intervals showing how much interaction-related heterogeneity remains in the underlying local effects (see, e.g., Figures 11). For ALE, a separate plot visualizes the interaction-related heterogeneity via the standard deviation of local effects, which is inspired by the derivative ICE plots of Goldstein et al. (2015) (see, e.g., Figure 12). For SD, the Shapley values that were recalculated conditioned on each subspace \mathcal{A}_g are visualized with the regional effect curve (see, e.g., Figure 18).

Note that we can also visualize the non-centered PD ($\hat{f}_{j|\mathcal{A}_g}^{PD}$) instead of the mean-centered PD, which might provide more insights regarding interpretation. However, the interaction-related heterogeneity must be estimated by the mean-centered ICE curves to only represent heterogeneity induced by feature interactions (see Appendix B.4).

C.4 Handling of Categorical Features

In this section, we will summarize the particularities of categorical features. Compared to numeric features, we have a limited number of K values (categories). Hence, compared to numeric features, we find split points by dividing the K categories into the two new

subspaces. Since GADGET is based on the general concept of a CART (decision tree) algorithm (Breiman et al., 1984) that can handle categorical features, the splitting itself follows the same approach as for a common CART algorithm. If we have categorical features in S , it does influence the calculation of our objective function, and the handling depends on the underlying feature effect method. We briefly discuss the specifics for each of the three feature effect methods that we use in this paper. All of them are able to handle categorical features, which is a general requirement that we can use it for mixed data sets within GADGET.

PD Plot. Compared to numeric features, the grid points for categorical features are limited to the number of categories. Otherwise, the calculation of the loss and the risk (see, Eq. 13 and Eq. 11) works exactly the same as for numeric features.²³

ALE Plot. ALE builds intervals based on quantiles for numeric features to calculate prediction differences between neighboring interval borders for all observations falling within this interval. For binary features, the authors solve this as follows: for all observations falling in each of the categories, the prediction difference when changing it to the other category is calculated. For more categories, they suggest a sorting algorithm.²⁴ Hence, we still receive the needed derivatives for GADGET for each category to calculate the loss and risk function for GADGET.

SD Plot. Compared to numeric features, the x-axis of the SD plot is a grid of size K . For each of these grid points (categories), the Shapley values for the observations belonging to the specific category are calculated. Hence, instead of a spline to quantify the expected value in Eq. (17), we use the arithmetic mean within each category (similar to PD plots) and, thus, sum up the variance of Shapley values for each category over all categories (within the respective subspace).

In general, if we apply GADGET and split w.r.t. a categorical feature such that only one category is present within a subspace (e.g., we split the feature sex such that all individuals are male in one resulting subspace), then the interaction related heterogeneity vanishes to zero, since only an additive shift for the feature sex is left in this subspace.

Note: If a categorical feature \mathbf{x}_j is not only considered for splitting ($j \in Z$) but is also a feature of interest ($j \in S$), the different splitting possibilities of categories of \mathbf{x}_j prompt recalculation for ALE, since derivatives are only calculated for pre-sorted neighboring categories. In our implementations, we only split w.r.t. the pre-sorted categories and considered them as integer values to reduce the computational burden of the calculations.

Appendix D. Additions to PINT

Although the approximation of the null distribution via theoretical distributions reduces the computational burden when applying PINT (as described in Section 5), it is still high for high-dimensional settings. Therefore, we suggest to randomly select a smaller set of observations to apply PINT in the case of a high number of observations. In the case of

23. Since we calculate the loss point-wise at each grid point and sum it up over all grid points, the order of the category does not make a difference for the objective of GADGET.

24. For more information, we refer to Apley and Zhu (2020).

DECOMPOSING GLOBAL FEATURE EFFECTS

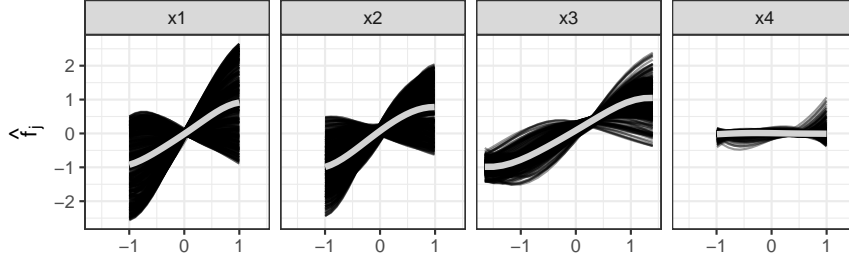


Figure 16: Mean-centered ice curves (black) and mean-centered PD curves (grey) for all four features of one repetition of the simulation example described in Section 5.

many features, we recommend to filter features beforehand and apply PINT only once to those with the highest possibility of containing feature interactions. For example, this can be done by excluding features that show only small variations of the local feature effects used in GADGET, since the homogeneous feature effects are a strong indicator that the feature does not interact with any other feature. For instance, the mean-centered ICE curves of \mathbf{x}_4 in Figure 16 show only small variation, which indicates that \mathbf{x}_4 does not interact with any other feature, and thus there is no need to consider it for PINT. Hence, we can apply PINT only on the remaining three features to identify which features interact with each other and must be considered within GADGET.

Appendix E. Further Details on Real-World Applications

COMPAS Data Set. Our analysis in Section 7 for the COMPAS data set is based on a tuned SVM. We chose this model based on the following selection process: For the binary classification task²⁵, we chose to select the best model out of a logistic regression, a random forest, and a tuned SVM with RBF kernel. We also compared these models to a featureless model. The model selection was performed by a 5-fold cross-validation, where the hyperparameters cost C and inverse kernel width σ of the SVM were tuned via 3-fold cross-validation on the inner folds of the nested resampling approach.²⁶ We evaluate the learners' test performance on the outer folds based on the F1 score and Matthews correlation coefficient (MCC) (Chicco and Jurman, 2020). Since the tuned SVM performs best w.r.t. both evaluation metrics, we chose this model for our further analysis. Note that the performance differences between the different learners (besides the featureless baseline) are very small. Thus, one might consider using the most interpretable learner (here, logistic regression) for further analysis. However, since the purpose of our analysis is to detect feature

25. The classes are slightly unbalanced, with 1317 defendants who have a high risk of recidivism and 2056 defendants with a low risk of recidivism.

26. For tuning, we used random search with 30 iterations on a search space of 2^{-12} to 2^{12} for each of the two hyperparameters.

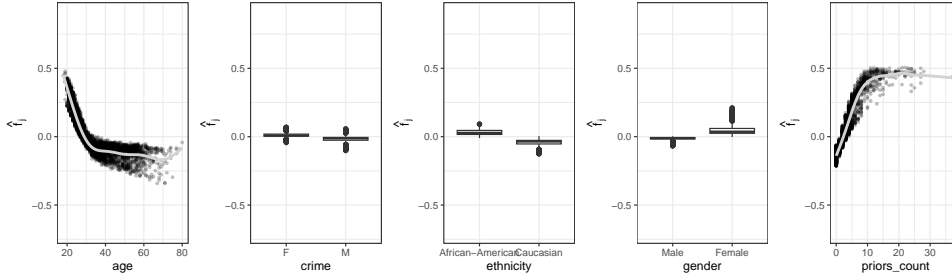


Figure 17: Global SD curves and Shapley values of considered features of the COMPAS application example.

interactions and reduce interaction-related heterogeneity by partitioning the feature space, here we choose the best-performing model that also potentially learned feature interactions.

Learner	MCC	F1 Score
Featureless	0.0000	0.7573
Logistic Regression	0.4675	0.8057
Random Forest	0.4710	0.8125
Support Vector Machine (tuned)	0.4752	0.8126

Table 6: Average learner test performance on 5-fold cross-validation for COMPAS data set.

In addition to the results of GADGET based on PD presented in Section 7, we also applied GADGET with the same settings based on SD.

The effect plots for the four resulting regions are shown in Figure 18. GADGET based on SD performs the same first split as for PD. The second split is also executed according to the number of prior crimes, but the split value is lower than for PD (at 2.5 instead of 4.5). The total interaction-related heterogeneity reduction ($R_{Tot}^2 = 0.87$) is also similar to that when PD is used. Note that Shapley values explain the difference between the the actual and average prediction. Hence, SD plots are centered, while Figure 11 shows the uncentered regional PD plots.

Bikesharing Data Set. The results in Section 7 for the Bikesharing data set are based on a random forest. We chose this model based on the following selection process: For the underlying regression task, we selected the best-performing model out of a linear model, a random forest, and a tuned SVM with RBF kernel. As a baseline comparison, we also report the performance of a featureless model. The model selection was performed by a 5-fold cross-validation, where the hyperparameters cost C and inverse kernel width σ of the SVM were tuned via 3-fold cross-validation on the inner folds of the nested resampling

DECOMPOSING GLOBAL FEATURE EFFECTS

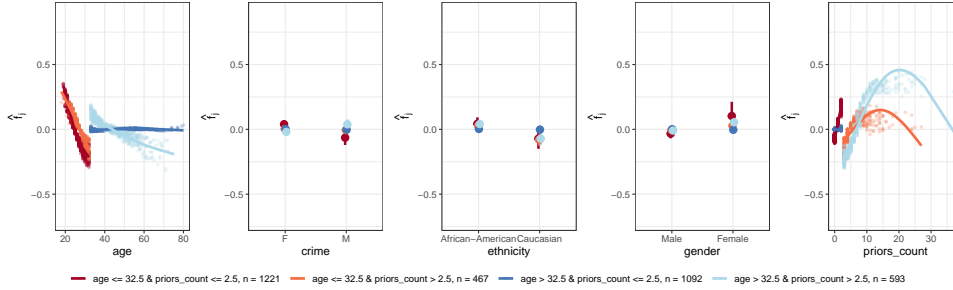


Figure 18: Regional SD plots for considered features of the COMPAS application after applying GADGET. Shapley values within each region are recalculated and visualize the interaction-related heterogeneity within each region.

approach.²⁷ We evaluate the learners’ test performance on the outer folds based on the MSE and R^2 . The random forest performed best and, hence, was chosen for further analyses.

Learner	MSE	R^2
Featureless	17902	-0.0002
Linear Regression	6641	0.6289
Random Forest	1077	0.9397
Support Vector Machine (tuned)	3365	0.8114

Table 7: Average learner test performance on 5-fold cross-validation for bikesharing data set.

References

Kjersti Aas, Martin Jullum, and Anders Løland. Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artificial Intelligence*, 298:103502, September 2021.

André Altmann, Laura Toloşi, Oliver Sander, and Thomas Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347, 04 2010.

Daniel W. Apley and Jingyu Zhu. 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, 2020.

²⁷ For tuning, we used random search with 30 iterations on a search space of 2^{-12} to 2^{12} for each of the two hyperparameters.

-
- Irene V Blair, Charles M Judd, and Kristine M Chapleau. The influence of afrocentric facial features in criminal sentencing. *Psychological science*, 15(10):674–679, 2004.
- Stefan Blücher, Johanna Vielhaben, and Nils Strodthoff. Preddiff: Explanations and interactions from conditional expectations. *Artificial Intelligence*, 312:103774, November 2022.
- Sebastian Bordt and Ulrike von Luxburg. From Shapley values to generalized additive models and back. In *International Conference on Artificial Intelligence and Statistics*, pages 709–745. PMLR, 2023.
- Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.
- Matthew Britton. Vine: Visualizing statistical interactions in black box models. *arXiv preprint arXiv:1904.00561*, 2019.
- Siu Lun Chau, Robert Hu, Javier Gonzalez, and Dino Sejdinovic. Rkhs-shap: Shapley values for kernel methods. In *Advances in Neural Information Processing Systems*, volume 35, pages 13050–13063, 2022.
- Hugh Chen, Joseph D. Janizek, Scott Lundberg, and Su-In Lee. True to the model or true to the data? *arXiv preprint arXiv:2006.16234*, 2020.
- Davide Chicco and Giuseppe Jurman. The advantages of the Matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21:1–13, 2020.
- Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable’s importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177):1–81, 2019.
- Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.
- Jerome H. Friedman and Bogdan E. Popescu. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3):916–954, 2008.
- Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, 24(1):44–65, January 2015.
- Brandon M. Greenwell, Bradley C. Boehmke, and Andrew J. McCarthy. A simple and effective model-based variable importance measure. *arXiv preprint arXiv:1805.04755*, 2018.
- Julia Herbinger, Bernd Bischl, and Giuseppe Casalicchio. Repid: Regional effect plots with implicit interaction detection. In *International Conference on Artificial Intelligence and Statistics*, pages 10209–10233. PMLR, 2022.

DECOMPOSING GLOBAL FEATURE EFFECTS

- Andrew Herren and P. Richard Hahn. Statistical aspects of shap: Functional anova for model interpretation. *arXiv preprint arXiv:2208.09970*, 2022.
- Munir Hiabu, Joseph T Meyer, and Marvin N Wright. Unifying local and global model explanations by functional decomposition of low dimensional structures. In *International Conference on Artificial Intelligence and Statistics*, pages 7040–7060. PMLR, 2023.
- Giles Hooker. Discovering additive structure in black box functions. In *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 575–580, 2004.
- Giles Hooker. Generalized functional anova diagnostics for high-dimensional functions of dependent variables. *Journal of Computational and Graphical Statistics*, 16(3):709–732, September 2007.
- Linwei Hu, Jie Chen, Vijayan N Nair, and Agus Sudjianto. Surrogate locally-interpretable models with supervised machine learning algorithms. *arXiv preprint arXiv:2007.14528*, 2020.
- Linwei Hu, Jie Chen, and Vijayan N. Nair. Using model-based trees with boosting to fit low-order functional anova models. *arXiv preprint arXiv:2207.06950*, 2022.
- Alan Inglis, Andrew Parnell, and Catherine B. Hurley. Visualizing variable importance and variable interaction effects in machine learning models. *Journal of Computational and Graphical Statistics*, 31(3):766–778, July 2022.
- Gareth James, Daniela Witten, Trevor Hastie, and Rob Tibshirani. *ISLR2: Introduction to Statistical Learning, Second Edition*, 2022. R package version 1.3-2.
- Neil Jethani, Mukund Sudarshan, Ian Connick Covert, Su-In Lee, and Rajesh Ranganath. Fastshap: Real-time Shapley value estimation. In *International Conference on Learning Representations*, 2021.
- Indra Kumar, Carlos Scheidegger, Suresh Venkatasubramanian, and Sorelle Friedler. Shapley residuals: Quantifying the limits of the Shapley value for explanations. In *Advances in Neural Information Processing Systems*, volume 34, pages 26598–26608, 2021.
- Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. How we analyzed the compas recidivism algorithm. *ProPublica*, 2016. URL <https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm>.
- Benjamin Lengerich, Sarah Tan, Chun-Hao Chang, Giles Hooker, and Rich Caruana. Purifying interaction effects with the functional anova: An efficient algorithm for recovering identifiable additive models. In *International Conference on Artificial Intelligence and Statistics*, pages 2402–2412. PMLR, 2020.
- Genyuan Li and Herschel Rabitz. General formulation of hdmr component functions with independent and correlated variables. *Journal of Mathematical Chemistry*, 50(1):99–130, January 2012.

- Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.
- Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30, 2017.
- Scott M. Lundberg, Gabriel G. Erion, and Su-In Lee. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*, March 2019.
- Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, 2(1): 56–67, 2020.
- Aria Masoomi, Davin Hill, Zhonghui Xu, Craig P Hersh, Edwin K Silverman, Peter J Castaldi, Stratis Ioannidis, and Jennifer Dy. Explanations of black-box models based on directional feature interactions. *arXiv preprint arXiv:2304.07670*, 2023.
- Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.
- Christoph Molnar. *Interpretable Machine Learning*. 2 edition, 2022. URL <https://christophm.github.io/interpretable-ml-book>.
- Christoph Molnar, Gunnar König, Julia Herbinger, Timo Freiesleben, Susanne Dandl, Christian A Scholbeck, Giuseppe Casalicchio, Moritz Grosse-Wentrup, and Bernd Bischl. General pitfalls of model-agnostic interpretation methods for machine learning models. In *xxAI-Beyond Explainable AI: International Workshop, Held in Conjunction with ICML 2020, July 18, 2020, Vienna, Austria, Revised and Extended Papers*, pages 39–68. Springer, 2022.
- Christoph Molnar, Gunnar König, Bernd Bischl, and Giuseppe Casalicchio. Model-agnostic feature importance and effects with dependent features: a conditional subgroup approach. *Data Mining and Knowledge Discovery*, pages 1–39, 2023.
- Sharif Rahman. A generalized anova dimensional decomposition for dependent probability measures. *SIAM/ASA Journal on Uncertainty Quantification*, 2:670–697, January 2014.
- Christian A Scholbeck, Giuseppe Casalicchio, Christoph Molnar, Bernd Bischl, and Christian Heumann. Marginal effects for non-linear prediction functions. *arXiv preprint arXiv:2201.08837*, 2022.
- Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28): 307–317, 1953.
- Margaret A Shipp, Ken N Ross, Pablo Tamayo, Andrew P Weng, Jeffery L Kutok, Ricardo CT Aguiar, Michelle Gaasenbeek, Michael Angelo, Michael Reich, Geraldine S Pinkus, et al. Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature medicine*, 8(1):68–74, 2002.

DECOMPOSING GLOBAL FEATURE EFFECTS

- Daria Sorokina, Rich Caruana, Mirek Riedewald, and Daniel Fink. Detecting statistical interactions with additive groves of trees. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1000–1007, 2008.
- Clemens Stachl, Florian Pargent, Sven Hilbert, Gabriella M Harari, Ramona Schoedel, Sumer Vaid, Samuel D Gosling, and Markus Bühner. Personality research and assessment in the era of machine learning. *European Journal of Personality*, 34(5):613–631, 2020.
- Charles J. Stone. The use of polynomial splines and their tensor products in multivariate function estimation. *The Annals of Statistics*, 22(1):118 – 171, 1994.
- Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, 41:647–665, 2014.
- Xingzhi Sun, Ziyu Wang, Rui Ding, Shi Han, and Dongmei Zhang. Puregam: Learning an inherently pure additive model. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1728–1738, 2022.
- Mukund Sundararajan and Amir Najmi. The many Shapley values for model explanation. In *Proceedings of the 37th International Conference on Machine Learning*, pages 9269–9278, 2020.
- Sarah Tan, Giles Hooker, Paul Koch, Albert Gordo, and Rich Caruana. Considerations when learning additive explanations for black-box models. *arXiv preprint arXiv:1801.08640*, 2018.
- Che-Ping Tsai, Chih-Kuan Yeh, and Pradeep Ravikumar. Faith-shap: The faithful Shapley interaction index. *Journal of Machine Learning Research*, 24(94):1–42, 2023.
- Xiaohang Zhang, Yuan Wang, and Zhengren Li. Interpreting the black box of supervised learning models: Visualizing the impacts of features on prediction. *Applied Intelligence*, 51(10):7151–7165, October 2021.

7. Explaining Hyperparameter Optimization via Partial Dependence Plots

The third contributing article of Part III also addresses the second limitation stated in Section 1.1. However, compared to the articles in Sections 5 and 6, we analyze the aggregation bias of global effect methods caused by extrapolation in this work. Therefore, this article focuses on explaining hyperparameter effects with PD plots in the context of automated ML. Hence, if efficient optimizers, such as Bayesian optimization, are used for hyperparameter optimization, we obtain regions in the hyperparameter space that are sparsely sampled, leading to uncertain predictions in these regions and, thus, to unreliable PD estimates. Therefore, we suggest a recursive partitioning algorithm, which partitions the hyperparameter space such that regional PD estimates are more reliable and confident in relevant regions.

Contributing article: Moosbauer, J., Herbinger, J., Casalicchio, G., Lindauer, M., and Bischl, B. (2021). Explaining hyperparameter optimization via partial dependence plots. In *Advances in Neural Information Processing Systems*, Volume 34, pp. 2280–2291.

Author contributions: Julia Herbinger and Julia Moosbauer share the first authorship of this paper. Their overall equal contributions can be described as follows:

Julia Moosbauer and Julia Herbinger developed the project idea with continuous support from Giuseppe Casalicchio. Julia Moosbauer has developed the idea of leveraging an uncertainty estimate to improve interpretability measures and derived the uncertainty estimate for partial dependent plots. Julia Herbinger has formed the initial core idea for the partitioning method to identify subregions based on uncertainty estimates with support from Giuseppe Casalicchio. The algorithm was developed by Julia Moosbauer based on an initial tree-splitting algorithm implemented by Giuseppe Casalicchio and was substantially improved by Julia Herbinger. Julia Moosbauer and Julia Herbinger jointly designed and conducted the experiments. Evaluation metrics for the benchmark were defined by Julia Herbinger and improved by Julia Moosbauer. Julia Moosbauer implemented the benchmark on synthetic functions, and Julia Herbinger implemented the deep learning benchmark. The manuscript was drafted jointly by Julia Moosbauer and Julia Herbinger with overall equal contributions. All authors contributed to revisions of the paper. Giuseppe Casalicchio, Marius Lindauer, and Bernd Bischl gave valuable input throughout the project and suggested several notable modifications.

Supplementary material available at: https://proceedings.neurips.cc/paper_files/paper/2021/file/12ced2db6f0193dda91ba86224ea1cd8-Supplemental.pdf

Explaining Hyperparameter Optimization via Partial Dependence Plots

Julia Moosbauer*, **Julia Herbinger***, **Giuseppe Casalicchio**, **Marius Lindauer**, **Bernd Bischl**
Department of Statistics, Ludwig-Maximilians-University Munich, Munich, Germany
Institute of Information Processing, Leibniz University Hannover, Hannover, Germany
{julia.moosbauer, julia.herbinger, giuseppe.casalicchio,
bernd.bischl}@stat.uni-muenchen.de
lindauer@tnt.uni-hannover.de

Abstract

Automated hyperparameter optimization (HPO) can support practitioners to obtain peak performance in machine learning models. However, there is often a lack of valuable insights into the effects of different hyperparameters on the final model performance. This lack of explainability makes it difficult to trust and understand the automated HPO process and its results. We suggest using interpretable machine learning (IML) to gain insights from the experimental data obtained during HPO with Bayesian optimization (BO). BO tends to focus on promising regions with potential high-performance configurations and thus induces a sampling bias. Hence, many IML techniques, such as the *partial dependence plot* (PDP), carry the risk of generating biased interpretations. By leveraging the posterior uncertainty of the BO surrogate model, we introduce a variant of the PDP with estimated confidence bands. We propose to partition the hyperparameter space to obtain more confident and reliable PDPs in relevant sub-regions. In an experimental study, we provide quantitative evidence for the increased quality of the PDPs within sub-regions.

1 Introduction

Most machine learning (ML) algorithms are highly configurable. Their hyperparameters must be chosen carefully, as their choice often impacts the model performance. Even for experts, it can be challenging to find well-performing hyperparameter configurations. Automated machine learning (AutoML) systems and methods for automated HPO have been shown to yield considerable efficiency compared to manual tuning by human experts [Snoek et al., 2012]. However, these approaches mainly return a well-performing configuration and leave users without insights into decisions of the optimization process. Questions about the importance of hyperparameters or their effects on the resulting performance often remain unanswered. Not all data scientists trust the outcome of an AutoML system due to the lack of transparency [Drozdal et al., 2020]. Consequently, they might not deploy an AutoML model, despite all performance gains. Providing insights into the search process may help increase trust and facilitate interactive and exploratory processes: A data scientist could monitor the AutoML process and make changes to it (e.g., restricting or expanding the search space) already *during* optimization to anticipate unintended results.

Transparency, trust, and understanding of the inner workings of an AutoML system can be increased by interpreting the internal surrogate model of an AutoML approach. For example, BO trains a surrogate model to approximate the relationship between hyperparameter configurations and model performance. It is used to guide the optimization process towards the most promising regions of the hyperparameter space. Hence, surrogate models implicitly contain information about the influence of

*These authors contributed equally to this work.

hyperparameters. If the interpretation of the surrogate matches with a data scientist’s expectation, confidence in the correct functioning of the system may be increased. If these do not match, it provides an opportunity to look either for bugs in the code or for new theoretical insights.

We propose to analyze surrogate models with methods from IML to provide insights into the results of HPO. In the context of BO, typical choices for surrogate models are flexible, probabilistic black-box models, such as Gaussian processes (GP) or random forests. Interpreting the effect of single hyperparameters on the performance of the model to be tuned is analogous to interpreting the feature effect of the black-box surrogate model. We focus on the PDP [Friedman, 2001], which is a widely-used method² to visualize the average marginal effect of single features on a black-box model’s prediction. When applied to surrogate models, they provide information on how a specific hyperparameter influences the estimated model performance. However, applying PDPs out of the box to surrogate models might lead to misleading conclusions. Efficient optimizers such as BO tend to focus on exploiting promising regions of the hyperparameter space while leaving other regions less explored. Therefore, a sampling bias in input space is introduced, which in turn can lead to a poor fit and biased interpretations in underexplored regions of the space.

Contributions: We study the problem of sampling bias in experimental data produced by AutoML systems and the resulting bias of the surrogate model and assess its implications on PDPs. We then derive an uncertainty measure for PDPs of probabilistic surrogate models. In addition, we propose a method that splits the hyperparameter space into interpretable sub-regions of varying uncertainty to obtain sub-regions with more reliable and confident PDP estimates. In the context of BO, we provide evidence for the usefulness of our proposed methods on a synthetic function and in an experimental study in which we optimize the architecture and hyperparameters of a deep neural network. Our Supplementary Material provides (A) more background related to uncertainty estimates, (B) notes on how our methods are applied to hierarchical hyperparameter spaces, (C) details on the experimental setup and more detailed results, (D) a link to the source code.

Reproducibility and Open Science: The implementation of the proposed methods as well as reproducible scripts for the experimental analysis are provided in a public git-repository³.

2 Background and Related Work

Recent research has begun to question whether the evaluation of an AutoML system should be purely based on the generated models’ predictive performance without considering interpretability [Hutter et al., 2014a, Pfisterer et al., 2019, Freitas, 2019, Xanthopoulos et al., 2020]. Interpreting AutoML systems can be categorized as (1) interpreting the resulting ML model on the underlying dataset, or (2) interpreting the HPO process itself. In this paper, we focus on the latter.

Let $c : \Lambda \rightarrow \mathbb{R}$ be a *black-box* cost function, mapping a hyperparameter configuration $\lambda = (\lambda_1, \dots, \lambda_d)$ to the model error⁴ obtained by a learning algorithm with configuration λ . The hyperparameter space may be mixed, containing categorical and continuous hyperparameters. The goal of HPO is to find $\lambda^* \in \arg \min_{\lambda \in \Lambda} c(\lambda)$. Throughout the paper, we assume that a surrogate model $\hat{c} : \Lambda \rightarrow \mathbb{R}$ is given as an approximation to c . If the surrogate is assumed to be a GP, $\hat{c}(\lambda)$ is a random variable following a Gaussian posterior distribution. In particular, for any finite indexed family of hyperparameter configurations $(\lambda^{(1)}, \dots, \lambda^{(k)}) \in \Lambda^k$, the vector of estimated performance values is Gaussian with a posterior mean $\hat{m} = (\hat{m}(\lambda^{(i)}))_{i=1, \dots, k}$ and covariance $\hat{K} = (\hat{k}(\lambda^{(i)}, \lambda^{(j)}))_{i, j=1, \dots, k}$.

Hyperparameter Importance. Understanding which hyperparameters influence model performance can provide valuable insights into the tuning strategy [Probst et al., 2019]. To quantify relevance of hyperparameters, models that inherently quantify feature relevance – e.g., GPs with ARD kernel [Neil, 1996] – can be used as surrogate models. Hutter et al. [2014a] quantified the importance of hyperparameters based on a random forest fitted on data generated by BO, for which the importance of both the main and the interaction effects of hyperparameters was calculated by a functional ANOVA approach. Similarly, Sharma et al. [2019] quantified the hyperparameter importance of

²There exist various implementations [Greenwell, 2017, Pedregosa et al., 2011]), extensions [Greenwell et al., 2018, Goldstein et al., 2015] and applications [Friedman and Meulman, 2003, Cutler et al., 2007].

³https://github.com/slds-lmu/paper_2021_xautoml

⁴Typically, the model error is estimated via cross-validation or hold-out testing.

residual neural networks. These works highlight how useful it is to quantify the importance of hyperparameters. However, importance scores do not show *how* a specific hyperparameter affects the model performance according to the surrogate model. Therefore, we propose to visualize the assumed marginal effect of a hyperparameter. A model-agnostic interpretation method that can be used for this purpose is the PDP.

PDPs for Hyperparameters. Let $S \subset \{1, 2, \dots, d\}$ denote an index set of features, and let $C = \{1, 2, \dots, d\} \setminus S$ be its complement. The partial dependence (PD) function [Friedman, 2001] of $c: \Lambda \rightarrow \mathbb{R}$ for hyperparameter(s) S is defined as⁵

$$c_S(\lambda_S) := \mathbb{E}_{\lambda_C} [c(\lambda)] = \int_{\Lambda_C} c(\lambda_S, \lambda_C) d\mathbb{P}(\lambda_C). \quad (1)$$

When analyzing the PDP of hyperparameters, we are usually interested in how their values λ_S impact model performance uniformly across the hyperparameter space. In line with prior work [Hutter et al., 2014a], we therefore assume \mathbb{P} to be the uniform distribution over Λ_C . Computing $c_S(\lambda_S)$ exactly is usually not possible because c is unknown and expensive to evaluate in the context of HPO. Thus, the posterior mean \hat{m} of the probabilistic surrogate model $\hat{c}(\lambda)$ is commonly used as a proxy for c . Furthermore, the integral may not be analytically tractable for arbitrary surrogate models \hat{c} . Hence, the integral is approximated by Monte Carlo integration, i.e.,

$$\hat{c}_S(\lambda_S) = \frac{1}{n} \sum_{i=1}^n \hat{m}(\lambda_S, \lambda_C^{(i)}) \quad (2)$$

for a sample $(\lambda_C^{(i)})_{i=1, \dots, n} \sim \mathbb{P}(\lambda_C)$. $\hat{m}(\lambda_S, \lambda_C^{(i)})$ represents the marginal effect of λ_S for one specific instance i . Individual conditional expectation (ICE) curves [Goldstein et al., 2015] visualize the marginal effect of the i -th observation by plotting the value of $\hat{m}(\lambda_S, \lambda_C^{(i)})$ against λ_S for a set of grid points⁶ $\lambda_S^{(g)} \in \Lambda_S, g \in \{1, \dots, G\}$. Analogously, the PDP visualizes $\hat{c}_S(\lambda_S)$ against the grid points. Following from Eq. 2, the PDP visualizes the average over all ICE curves. In HPO, the marginal predicted performance is a related concept. Instead of approximating the integral via Monte Carlo, the integral over \hat{c} is computed exactly. Hutter et al. [2014a] propose an efficient approach to compute this integral for random forest surrogate models.

Uncertainty Quantification in PDPs. Quantifying the uncertainty of PDPs provides additional information about the reliability of the mean estimator. Hutter et al. [2014a] quantified the model uncertainty specifically for random forests as surrogates in BO by calculating the standard deviation of the marginal predictions of the individual trees. However, this procedure is not applicable to general probabilistic surrogate models, such as the commonly used GP. There are approaches that quantify the uncertainty for ML models that do not provide uncertainty estimates out-of-the-box. Cafri and Bailey [2016] suggested a bootstrap approach for tree ensembles to quantify the uncertainties of effects based on PDPs. Another approach to quantify the uncertainty of PDPs is to leverage the ICE curves. For example, Greenwell [2017] implemented a method that marginalizes over the mean \pm standard deviation of the ICE curves for each grid point. However, this approach quantifies the underlying uncertainty of the data at hand rather than the model uncertainty, as explained in Appendix A.1. A model-agnostic estimate based on uncertainty estimates for probabilistic models is missing so far.

Subgroup PDPs. Recently, a new research direction concentrates on finding more reliable PDP estimates within subgroups of observations. Molnar et al. [2020] focused on problems in PDP estimation with correlated features. To that end, they apply transformation trees to find homogeneous subgroups and then visualize a PDP for each subgroup. Grömping [2020] looked at the same problem and also uses subgroup PDPs, where ICE curves are grouped regarding a correlated feature. Britton [2019] applied a clustering approach to group ICE curves to find interactions between features. However, none of these approaches aim at finding subgroups where reliable PDP estimates have low uncertainty. Additionally, to the best of our knowledge, nothing similar exists for analyzing experimental data created by HPO.

⁵To keep notation simple, we denote $c(\lambda)$ as a function of two arguments (λ_S, λ_C) to differentiate components in the index set S from those in the complement. The integral shall be understood as a multiple integral of c where $\lambda_j, j \in C$, are integrated out.

⁶Grid points are typically chosen as an equidistant grid or sampled from $\mathbb{P}(\lambda_S)$. The granularity G is chosen by the user. For categorical features, the granularity typically corresponds to the number of categories.

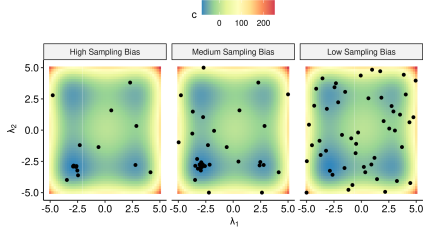


Figure 1: Illustration of the sampling bias when optimizing the 2D Styblinski Tang function with BO and the Lower Confidence Bound (LCB) acquisition function $a(\lambda) = \hat{m}(\lambda) + \tau \cdot \hat{s}(\lambda)$ for $\tau = 0.1$ (left) and $\tau = 2$ (middle) vs. data sampled uniformly at random (right).

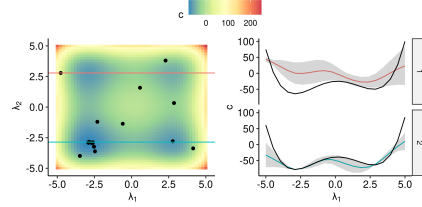


Figure 2: The two horizontal cuts (left) yield two ICE curves (right) showing the mean prediction and uncertainty band against λ_1 for \hat{c} with $\tau = 0.1$ on the 2D Styblinski-Tang function. The upper ICE curve deviates more from the true effect (black) and shows a higher uncertainty.

3 Biased Sampling in HPO

Visualizing the marginal effect of hyperparameters of surrogate models via PDPs can be misleading. We show that this problem is due to the sequential nature of BO, which generates dependent instances (i.e., hyperparameter configurations) and thereby introduces a sampling and a resulting model bias. To save computational resources in contrast to grid search or random search, efficient optimizers like BO tend to exploit promising regions of the hyperparameter space while other regions are less explored (see Figure 1). Consequently, predictions of surrogate models are usually more accurate with less uncertainty in well-explored regions and less accurate with high uncertainty in under-explored regions. This model bias also affects the PD estimate (see Figure 2). ICE curves may be biased and less confident if they are computed in poorly-learned regions where the model has not seen much data before. Under the assumption of uniformly distributed hyperparameters, poorly-learned regions are incorporated in the PD estimate with the same weight as well-learned regions. ICE curves belonging to regions with high uncertainty may obfuscate well-learned effects of ICE curves belonging to other regions when they are aggregated to a PDP. Hence, the model bias may also lead to a less reliable PD estimate. PDPs visualizing only the mean estimator of Eq. (2) do not provide insights into the reliability of the PD estimate and how it is affected by the described model bias.

4 Quantifying Uncertainty in PDPs

Pointwise uncertainty estimates of a probabilistic model provide insights into the reliability of the prediction $\hat{c}(\lambda)$ for a specific configuration λ . This uncertainty directly correlates with how explored the region around λ is. Hence, including the model's uncertainty structure into the PD estimate enables users to understand in which regions the PDP is more reliable and which parts of the PDP must be cautiously interpreted.⁷ We now extend the PDP of Eq. (2) to probabilistic surrogate models \hat{c} (e.g., a GP). Let λ_S be a fixed grid point and $(\lambda_C^{(i)})_{i=1,\dots,n} \sim \mathbb{P}(\lambda_C)$ a sample that is used to compute the Monte Carlo estimate of Eq. (2). The vector of predicted performances at the grid point λ_S is $\hat{c}(\lambda_S) = (\hat{c}(\lambda_S, \lambda_C^{(i)}))_{i=1,\dots,n}$ with (posterior) mean $\hat{m}(\lambda_S) := (\hat{m}(\lambda_S, \lambda_C^{(i)}))_{i=1,\dots,n}$ and

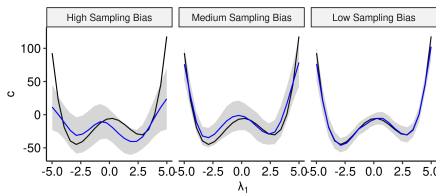


Figure 3: PDPs (blue) with confidence bands for surrogates trained on data created by BO and LCB with $\tau = 0.1$ (left), $\tau = 1$ (middle) and uniform i.i.d. dataset (right) vs. the true PD (black).

⁷Note that we aim at representing model uncertainty in a PD estimate, and not the variability of the mean prediction (see Appendix A.1 for a more detailed justification).

a (posterior) covariance $\hat{\mathbf{K}}(\boldsymbol{\lambda}_S) := \left(\hat{k} \left(\left(\boldsymbol{\lambda}_S, \boldsymbol{\lambda}_C^{(i)} \right), \left(\boldsymbol{\lambda}_S, \boldsymbol{\lambda}_C^{(j)} \right) \right) \right)_{i,j=1,\dots,n}$. Thus, $\hat{c}_S(\boldsymbol{\lambda}_S) = \frac{1}{n} \sum_{i=1}^n \hat{c}(\boldsymbol{\lambda}_S, \boldsymbol{\lambda}_C^{(i)})$ is a random variable itself. The expected value of $\hat{c}_S(\boldsymbol{\lambda}_S)$ corresponds to the PD of the posterior mean function \hat{m} at $\boldsymbol{\lambda}_S$, i.e.:

$$\hat{m}_S(\boldsymbol{\lambda}_S) = \mathbb{E}_{\hat{c}}[\hat{c}_S(\boldsymbol{\lambda}_S)] = \mathbb{E}_{\hat{c}} \left[\frac{1}{n} \sum_{i=1}^n \hat{c}(\boldsymbol{\lambda}_S, \boldsymbol{\lambda}_C^{(i)}) \right] = \frac{1}{n} \sum_{i=1}^n \hat{m}(\boldsymbol{\lambda}_S, \boldsymbol{\lambda}_C^{(i)}). \quad (3)$$

The variance of $\hat{c}_S(\boldsymbol{\lambda}_S)$ is

$$\hat{s}_S^2(\boldsymbol{\lambda}_S) = \mathbb{V}_{\hat{c}}[\hat{c}_S(\boldsymbol{\lambda}_S)] = \mathbb{V}_{\hat{c}} \left[\frac{1}{n} \sum_{i=1}^n \hat{c}(\boldsymbol{\lambda}_S, \boldsymbol{\lambda}_C^{(i)}) \right] = \frac{1}{n^2} \mathbf{1}^\top \hat{\mathbf{K}}(\boldsymbol{\lambda}_S) \mathbf{1}. \quad (4)$$

For the above estimate, it is important that the kernel is correctly specified such that the covariance structure is modeled properly by the surrogate model. Eq. (4) can be approximated empirically by treating the pairwise covariances as unknown, i.e.:

$$\hat{s}_S^2(\boldsymbol{\lambda}_S) \approx \frac{1}{n} \sum_{i=1}^n \hat{\mathbf{K}}(\boldsymbol{\lambda}_S)_{i,i}. \quad (5)$$

In Appendix A.2, we show empirically that this approximation is less sensitive to kernel misspecifications. Please note that the variance estimate and the mean estimate can also be applied to other probabilistic models, such as GAMLSS⁸, transformation trees, or a random forest. An example for PDPs with uncertainty estimates is shown in Figure 3 for different degrees of a sampling bias.

5 Regional PDPs via Confidence Splitting

As discussed in Section 3, (efficient) optimization may imply that the sampling is biased, which in turn can produce misleading interpretations when IML is naively applied. We now aim to identify sub-regions $\Lambda' \subset \Lambda$ of the hyperparameter space in which the PD can be estimated with high confidence, and separate those from sub-regions in which it cannot be estimated reliably. In particular, we identify sub-regions in which poorly-learned effects do not obfuscate the well-learned effects along each grid point, thereby allowing the user to draw conclusions with higher confidence. By partitioning the entire hyperparameter space through a tree-based approach into disjoint and interpretable sub-regions, a more detailed understanding of the sampling process and hyperparameter effects is achieved. Users can either study the hyperparameter effect of a (confident) sub-region individually or understand the exploration-exploitation sampling of HPO by considering the complete tree structure. The result of this procedure for a single split is shown in Figure 5.

The PD estimate on the *entire* hyperparameter space Λ is computed by sampling the Monte Carlo estimate $(\boldsymbol{\lambda}_C^{(i)})_{i \in \mathcal{N}} \sim \mathbb{P}(\boldsymbol{\lambda}_C)$, $\mathcal{N} := \{1, 2, \dots, n\}$. We now introduce the PD estimate on a *sub-region* $\Lambda' \subset \Lambda$ simply as $(\boldsymbol{\lambda}_C^{(i)})_{i \in \mathcal{N}'}$ only using $\mathcal{N}' = \{i \in \mathcal{N}\}_{\boldsymbol{\lambda}^{(i)} \in \Lambda'}$. Since we are interested in the marginal effect of the hyperparameter(s) S at each $\boldsymbol{\lambda}_S \in \Lambda_S$, we will usually visualize the PD for the whole range Λ_S . Thus, all obtained sub-regions should be of the form $\Lambda' = \Lambda_S \times \Lambda'_C$ with $\Lambda'_C \subset \Lambda_C$. This corresponds to an average of ICE curves in the set $i \in \mathcal{N}'$. The pseudo-code to partition a hyperparameter (sub-)space Λ and corresponding sample $(\boldsymbol{\lambda}_C^{(i)})_{i \in \mathcal{N}} \in \Lambda_C$, $\mathcal{N} \subseteq \{1, \dots, n\}$, into two child regions is shown in Algorithm 1. This splitting is recursively applied in a CART⁹-like procedure [Breiman et al., 1984b] to expand a full tree structure, with the usual stopping criteria (e.g., a maximum number of splits, a minimum size of a region, or a minimum improvement in each node). In each leaf node, the sub-regional PDP and its corresponding uncertainty estimate are computed by aggregating over all contained ICE curves.

The criterion to evaluate a specific partitioning is based on the idea of grouping ICE curves with similar uncertainty structure. To be more exact, we evaluate the impurity of a PD estimate on a sub-region Λ' with the help of the associated set of observations $\mathcal{N}' = \{i \in \mathcal{N}\}_{\boldsymbol{\lambda}_C^{(i)} \in \Lambda'_C}$, also referred to as nodes, as follows: For each grid point $\boldsymbol{\lambda}_S$, we use the L2 loss in $L(\boldsymbol{\lambda}_S, \mathcal{N}')$ to evaluate how the

⁸Generalized additive models for location, scale and shape

⁹Classification and regression trees

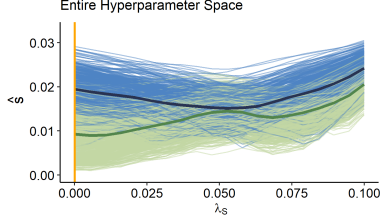


Figure 4: ICE curves of \hat{s} of λ_S for the left (green) and right (blue) sub-region after the first split. The darker lines represent the respective PDPs. The orange vertical line marks the value λ_S of the optimal configuration.

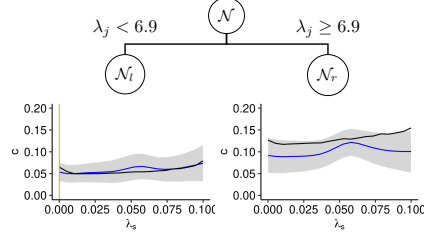


Figure 5: Example of two estimated PDPs (blue line) and 95% confidence bands after one partitioning step. The orange vertical line is the value of λ_S from the optimal configuration, the black curve is the true PD estimate $c_S(\lambda_S)$.

uncertainty varies across all ICE estimates $i \in \mathcal{N}'$ using $\hat{s}_{S|\mathcal{N}'}^2(\lambda_S) := \frac{1}{|\mathcal{N}'|} \sum_{i \in \mathcal{N}'} \hat{s}^2(\lambda_S, \lambda_C^{(i)})$ and aggregate the loss $\mathcal{L}(\lambda_S, \mathcal{N}')$ over all grid points in $\mathcal{R}_{L_2}(\mathcal{N}')$:

$$\mathcal{L}(\lambda_S, \mathcal{N}') = \sum_{i \in \mathcal{N}'} \left(\hat{s}^2(\lambda_S, \lambda_C^{(i)}) - \hat{s}_{S|\mathcal{N}'}^2(\lambda_S) \right)^2 \text{ and } \mathcal{R}_{L_2}(\mathcal{N}') = \sum_{g=1}^G \mathcal{L}(\lambda_S^{(g)}, \mathcal{N}'). \quad (6)$$

Algorithm 1: Tree-based Partitioning

input: \mathcal{N}
for $j \in C$ **do**
 for Every split t on hyperparameter λ_j **do**
 $\mathcal{N}_l^{j,t} = \{i \in \mathcal{N}\}_{\lambda_j^{(i)} \leq t}$
 $\mathcal{N}_r^{j,t} = \{i \in \mathcal{N}\}_{\lambda_j^{(i)} > t}$
 $\mathcal{I}(j, t) = \mathcal{R}_{L_2}(\mathcal{N}_l^{j,t}) + \mathcal{R}_{L_2}(\mathcal{N}_r^{j,t})$
 end for
 end for
 Choose $(j^*, t_{\lambda_j^*}^*) \in \arg \min_{j,t} \mathcal{I}(j, t)$
 Return $\mathcal{N}_l^{j^*, t_{\lambda_j^*}^*}$ and $\mathcal{N}_r^{j^*, t_{\lambda_j^*}^*}$ for $(j, t) = (j^*, t_{\lambda_j^*}^*)$

Hence, we measure the pointwise L_2 -distance between ICE curves of the variance function $\hat{s}^2(\lambda_S, \lambda_C^{(i)})$ and its PD estimate $\hat{s}_{S|\mathcal{N}'}^2(\lambda_S)$ within a sub-region \mathcal{N}' . This seems reasonable, as ICE curves in well-explored regions of the search space should, on average, have a lower uncertainty than those in less-explored regions. However, since we only split according to hyperparameters in C but not in S , the partitioning does not cut off less explored regions w.r.t. λ_S . Thus, the chosen split criterion groups ICE curves of the uncertainty estimate such that we receive sub-regions associated with low costs c (and thus high relevance for a user) to be more confident in well-explored regions of λ_S and less confident in under-explored regions. Figure 4 shows that ICE curves of the uncertainty measure with high uncertainty over the entire

range of λ_S are grouped together (right sub-region). Those with low uncertainty close to the optimal configuration of λ_S and increasing uncertainties for less suitable configurations are grouped together by curve similarities in the left sub-region. The respective PDPs are illustrated in Figure 5, where the confidence band in the left sub-region decreased compared to the confidence band of the global PDP especially for grid points close to the optimal value of λ_S . Hence, by grouping observations with similar ICE curves of the variance function, resulting sub-regional PDPs with confidence bands provide the user with the information of which sub-regions of Λ_C are well-explored and lead to more reliable PDP estimates. Furthermore, the user will know which ranges of λ_S can be interpreted reliably and which ones need to be regarded with caution.

To sum up, the splitting procedure provides interpretable, disjoint sub-regions of the hyperparameter space. Based on the defined impurity measure, PDPs with high reliability can be identified and analyzed. In particular, the method provides more confident and reliable estimates in the sub-region containing the optimal configuration. Which PDPs are most interesting to explore depends on the question the user would like to answer. If the main interest lies in understanding the optimization and exploring the sampling process, a user might want to keep the number of sub-regions relatively low by performing only a few partitioning steps. Subsequently, one would investigate the overall structure of the sub-regions and the individual sub-regional PDPs. If users are more interested in interpreting

hyperparameter effects only in the most relevant sub-region(s), they may want to split deeper and only look at sub-regions that are more confident than the global PDP.

Due to the nature of the splitting procedure, the PDP estimate on the entire hyperparameter space is a weighted average of the respective sub-regional PDPs. Hence, the global PDP estimate is decomposed into several sub-regional PDP estimates. Furthermore, note that the proposed method does not assume a numeric hyperparameter space, since the uncertainty estimates as well as ICE and PDP estimates can also be calculated for categorical features. Thus, it is applicable to problems with mixed spaces as long as a probabilistic surrogate model – and particularly its uncertainty estimates – are available. In Appendix B we describe how our method is applied to hierarchical hyperparameter spaces.

Since the proposed method is an instance of the CART algorithm, finding the optimal split for a categorical variable with q levels generally involves checking 2^q subsets. This becomes computationally infeasible for high values of q . It remains an open question for future work if this can be sped by an optimal procedure as in regression with L2 loss [Fisher, 1958] and binary classification [Breiman et al., 1984a] or by a clever heuristic as for multiclass classification Wright and König [2019].

6 Experimental Analysis

In this section, we validate the effectiveness of the introduced methods. We formulate two main hypotheses: First, experimental data affected by the sampling bias lead to biased surrogate models and thus to unreliable and misleading PDPs. Second, the proposed partitioning allows us to identify an interpretable sub-region (around the optimal configuration) that yields a more reliable and confident PDP estimate. In a first experiment, we apply our methods to BO runs on a synthetic function. In this controlled setup, we investigate the validity of our hypotheses with regards to problems of different dimensionality and different degrees of sampling bias. In a second experiment, we evaluate our PDP partitioning in the context of HPO for neural networks on a variety of tabular datasets.

We assess the sampling bias of the optimization design points by comparing their empirical distribution to a uniform distribution via Maximum Mean Discrepancy (MMD) [Gretton et al., 2012, Molnar et al., 2020], which is covered in more detail in the Appendix C.1. We measure the reliability of a PDP, i.e., the degree to which a user can rely on the estimate of the PD estimate, by comparing it to the true PD $c_S(\lambda_S)$ as defined in Eq. (1). More specifically, for every grid point $\lambda_S^{(g)}$, we compute the negative log-likelihood (NLL) of $c_S(\lambda_S)$ under the distribution of $\hat{c}_S(\lambda_S)$ pointwise for every grid point $\lambda_S^{(g)}$. The confidence of a PDP is illustrated by the width of its confidence bands $\hat{m}_S(\lambda_S) \pm q_{1-\alpha/2} \cdot \hat{s}_S(\lambda_S)$, with $q_{1-\alpha/2}$ denoting the $(1 - \alpha/2)$ -quantile of a standard normal distribution. We measure the confidence by assessing $\hat{s}_S(\lambda_S)$ pointwise for every grid point. In particular, we consider the mean confidence (MC) across all grid points $\frac{1}{G} \sum_{g=1}^G \hat{s}(\lambda_S^{(g)})$ as well as the confidence at the grid point closest to $\hat{\lambda}_S$ abbreviated by OC, with $\hat{\lambda}$ being the best configuration evaluated by the optimizer. To evaluate the performance of the confidence splitting, we report the above metrics on the sub-region that contains the best configuration evaluated by the optimizer, assuming that this region is of particular interest for a user of HPO. PDPs are computed with regards to single features for $G = 20$ equidistant grid points and $n = 1000$ Monte Carlo samples.

6.1 BO on a Synthetic Function

We consider the d -dimensional Styblinski-Tang function $c : [-5, 5]^d \rightarrow \mathbb{R}$, $\lambda \mapsto \frac{1}{2} \sum_{i=1}^d (\lambda_i^4 + 16\lambda_i^2 + 5\lambda_i)$ for $d \in \{3, 5, 8\}$. Since the PD is the same for each dimension i , we only present the effects of λ_1 . We performed BO with a GP surrogate model with a Matérn-3/2 kernel and the LCB acquisition function $a(\lambda) = \hat{m}(\lambda) + \tau \cdot \hat{s}(\lambda)$ with different values $\tau \in \{0.1, 1, 5\}$ to control the sampling bias. We compute the global PDP with confidence bands estimated according to Eq. (5) for the GP surrogate model \hat{c} that was fitted in the *last* iteration of BO. We ran Algorithm 1, and computed the PDP in the sub-region containing the optimal configuration. All computations were repeated 30 times. Further details on the setup are given in Appendix C.2.1.

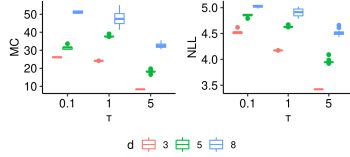


Figure 6: The figure presents the MC (left) and the NLL (right) for $d \in \{3, 5, 8\}$ for a high ($\tau = 0.1$), medium ($\tau = 1$), and low ($\tau = 5$) sampling bias across 30 replications. With a lower sampling bias, we obtain narrower confidence bands and a lower NLL.

Table 1: The table shows the relative improvement of the MC and the NLL via Algorithm 1 with 1 and 3 splits, compared to the global PDP along with the sampling bias for a $\tau = 0.1$ (high), $\tau = 2$ (medium), and $\tau = 5$ (low). Results are averaged across 30 replications.

d	MMD	δ MC (%)		δ NLL (%)	
		$n_{sp} = 1$	$n_{sp} = 3$	$n_{sp} = 1$	$n_{sp} = 3$
3	low (0.18)	7.65	13.64	5.89	10.92
3	medium (0.51)	12.86	36.92	4.78	7.70
3	high (0.56)	16.52	34.84	2.77	-1.62
5	low (0.15)	6.63	15.45	2.82	6.05
5	medium (0.45)	19.67	37.28	4.05	7.80
5	high (0.53)	11.99	33.06	-3.86	-1.93
8	low (0.11)	3.58	9.67	0.84	2.40
8	medium (0.42)	8.86	23.03	1.51	3.30
8	high (0.56)	6.59	19.84	1.53	4.29

As presented in Figure 6, the PDPs for surrogate models trained on *less biased* data (measured by the MMD) yield *lower* values of the NLL, as well as *lower* values for the MC. Table 1 shows that a single tree-based split reduces the MC by up to almost 20%, and up to 37% when performing 3 partitioning steps. Additionally, the NLL improves with an increasing number of partitioning steps in most cases. The results on the synthetic functions support our second hypothesis that the tree-based partitioning improves the reliability in terms of the NLL and the confidence of the PD estimates. The improvement of the MC is higher for a medium to high sampling bias, compared to scenarios that are less affected by sampling bias. We observe that (particularly for high sampling bias) there are some outlier cases in which the NLL worsens. More detailed results are shown in Appendix C.3.1.

6.2 HPO on Deep Learning

In a second experiment, we investigate HPO in the context of a surrogate benchmark [Eggenberger et al., 2015] based on the LCBench data [Zimmer et al., 2021]. For each of the 35 different OpenML [Vanschoren et al., 2013] classification tasks, LCBench provides access to evaluations of a deep neural network on 2000 configurations randomly drawn from the configuration space defined by Auto-PyTorch Tabular (see Table 5 in Appendix C.2). For each task, we trained a random forest as an empirical performance model that predicts the balanced validation error of the neural network for a given configuration. These empirical performance models serve as cheap to evaluate objective functions, which efficiently approximate the result of the real-world experiment of running a deep learning configuration on an LCBench instance. BO then acts on this empirical performance model as its objective¹⁰.

For each task, we ran BO to obtain the optimal architecture and hyperparameter configuration. Again, we used a GP with a Matérn-3/2 kernel and LCB with $\tau = 1$. Each BO run was allotted a budget of 200 objective function evaluations. We computed the PDPs and their confidences, which are estimated according to Eq. (5), based on the surrogate model \hat{c} after the final iteration. We performed tree-based partitioning with up to 6 splits based on a uniformly distributed dataset of size $n = 1000$. All computations were statistically repeated 30 times. Further details are provided in Appendix C.2.2.

For the real-world data example, we focus on answering the second hypothesis, i.e., whether the tree-based Algorithm 1 improves the reliability of the PD estimates. We compare the PDP in sub-regions after 6 splits with the global PDP. We computed the relative improvement of the confidence (MC and OC) and the NLL of the sub-regional PDP compared to the respective estimates for the global PDP. As shown in Table 2, the MC of the PDPs is on average reduced by 30% to 52%, depending on the hyperparameter. At the optimal configuration $\hat{\lambda}_S$, the improvement even increases to 50% – 62%. Thus, PDP estimates for all hyperparameters are on average – independent of the underlying dataset – clearly more confident in the relevant sub-regions when compared to the global PD estimates, especially around the optimal configuration $\hat{\lambda}_S$. In addition to the MC, the NLL simultaneously improves. In Appendix C.3.2, we provide details regarding the evaluated metrics on the level of the dataset and demonstrate that our split criterion outperforms other impurity measures regarding MC

¹⁰Please note that the random forest is only used as a surrogate in order to construct an efficient benchmark objective, and not as a surrogate in the BO algorithm, where we use a GP.

and OC. Furthermore, we emphasize in Appendix C.3.2 the significance of our results by providing a comparison to a naive baseline method.

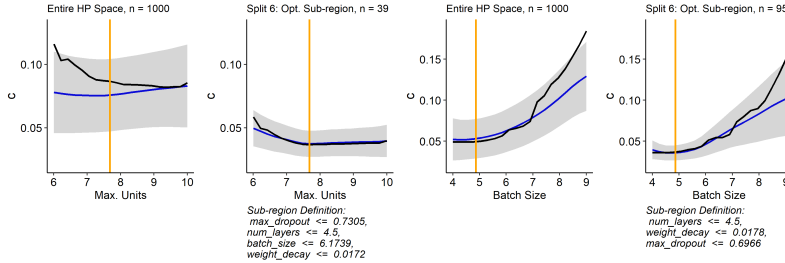


Figure 7: PDP (blue) and confidence band (grey) of the GP for hyperparameter *max. number of units* (*batch size*) on the left (right) side. The black line shows the PDP of the meta surrogate model representing the true PDP estimate. The orange vertical line marks the optimal configuration $\hat{\lambda}_S$. The relative improvements from the global PDP to the sub-regional PDP after 6 splits are for *max. number of units* (*batch size*): δ MC = 61.6% (28.4%), δ OC = 63.5% (62.2%), δ NLL = 48.6% (30.1%).

To further study our suggested method, we now highlight a few individual experiments. We chose one iteration of the *shuttle* dataset. On the two left plots of Figure 7, we see that the true PDP estimate for *max. number of units* is decreasing, while the globally estimated PDP trend is increasing and thus misleading. Although the confidence band already indicates that the PDP cannot be reliably interpreted on the entire hyperparameter space, it remains challenging to draw any conclusions from it. After performing 6 splits, we receive a confident and reliable PD estimate on an interpretable sub-region. The same plots are depicted for the hyperparameter *batch size* on the right part of Figure 7. This example illustrates that the confidence band might not always shrink uniformly over the entire range of λ_S during the partitioning, but often particularly around the optimal configuration $\hat{\lambda}_S$.

Table 2: Relative improvement of MC, OC, and NLL on hyperparameter level. The table shows the respective mean (standard deviation) of the average relative improvement over 30 replications for each dataset and 6 splits.

Hyperparameter	δ MC (%)	δ OC (%)	δ NLL (%)
Batch size	40.8 (14.9)	61.9 (13.5)	19.8 (19.5)
Learning rate	50.2 (13.7)	57.6 (14.4)	17.9 (20.5)
Max. dropout	49.7 (15.4)	62.4 (11.9)	17.4 (18.2)
Max. units	51.1 (15.2)	58.6 (12.7)	24.6 (22.0)
Momentum	51.7 (14.5)	58.3 (12.7)	19.7 (21.7)
Number of layers	30.6 (16.4)	50.9 (16.6)	13.8 (32.5)
Weight decay	36.3 (22.6)	61.0 (13.1)	11.9 (19.7)

7 Discussion and Conclusion

In this paper, we showed that partial dependence estimates for surrogate models fitted on experimental data generated by efficient hyperparameter optimization can be unreliable due to an underlying sampling bias. We extended PDPs by an uncertainty estimate to provide users with more information regarding the reliability of the mean estimator. Furthermore, we introduced a tree-based partitioning approach for PDPs, where we leverage the uncertainty estimator to decompose the hyperparameter space into interpretable, disjoint sub-regions. We showed with two experimental studies that we generate, on average, more confident and more reliable regional PDP estimates in the sub-region containing the optimal configuration compared to the global PDP.

One of the main limitations of PDPs is that they bear the risk of providing misleading results if applied to correlated data in the presence of interactions, especially for nonparametric models [Grömping, 2020]. However, existing alternatives that visualize the global marginal effect of a feature such as accumulated local effect (ALE) plots [Apley and Zhu, 2020] do also not provide a fully satisfying solution to this problem [Grömping, 2020]. As a solution to this problem, Grömping [2020] suggests stratified PDPs by conditioning on a correlated and potentially interacting feature to group ICE curves. This idea is in the spirit of our introduced tree-based partitioning algorithm. However, in the context of BO we might assume the distribution in Eq. (1) to be uniform and therefore no correlations are present. Instead of correlated features, we are faced with a sampling bias (see Section 3) where

we observe regions of varying uncertainty. Hence, instead of stratifying with respect to correlated features and aggregating ICE curves in regions with less correlated features, we stratify with respect to uncertainty and aggregate ICE curves in regions with low uncertainty variation. Nonetheless, it might be interesting to compare our approach with approaches based on the considerations made by Grömping [2020] – or potentially improved ALE curves.

Another limitation when using single-feature PDPs as in our examples is that hyperparameter interactions are not visible. However, two-way interactions can be visualized by plotting two-dimensional PDPs within sub-regions. Another possibility to detect interactions is to look at ICE curves within the sub-regions. If the shape of ICE curves within a sub-region is very heterogeneous, it indicates that the hyperparameter under consideration interacts with one of the other hyperparameters. Hence, having the additional possibility to look at ICE curves of individual observations within a sub-region is an advantage compared to other global feature effect plots such as ALE plots [Apley and Zhu, 2020], as they are not defined on an observational level. While we mainly discussed GP surrogate models on a numerical hyperparameter space in our examples, our methods are applicable to a wide variety of distributional regression models and also for mixed and hierarchical hyperparameter spaces. We also considered in Appendix C.3.2 different impurity measures. While the one introduced in this paper performed best in our experimental settings, this impurity measure as well as other components are exchangeable within the proposed algorithm. In the future, we will study our method on more complex, hierarchical configuration spaces for neural architecture search.

The proposed interpretation method is based on a surrogate and consequently does provide insights about what the AutoML system has *learned*, which in turn allows plausibility checks and may increase trust in the system. To what extent this allows conclusions on the *true* underlying hyperparameter effects depends on the quality of the surrogate. How to efficiently perform model diagnostics to ensure a high surrogate quality before applying interpretability techniques is subject to future research.

While we focused on providing better explanations without generating any additional experimental data, it might be interesting to investigate in future work how confidence and reliability of IML methods can be increased most efficiently when a user is allowed to conduct additional experiments.

Overall, we believe that increasing interpretability of AutoML will pave the way for human-centered AutoML. Our vision is that users will be able to better understand the reasoning and the sampling process of AutoML systems and thus can either trust and accept the results of the AutoML system or interact with it in a feedback loop based on the gained insights and their preferences. How users can then best interact with AutoML (beyond simple changes of the configuration space) will be left open for future research.

Acknowledgments and Disclosure of Funding

This work has been partially supported by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A. The authors of this work take full responsibilities for its content.

References

- D. W. Apley and J. Zhu. 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, 2020.
- B. Bischl, J. Richter, J. Bossek, D. Horn, J. Thomas, and M. Lang. mlrmo: A modular framework for model-based optimization of expensive black-box functions, 2018.
- L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984a. ISBN 0-534-98053-8.
- L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984b.
- M. Britton. VINE: Visualizing statistical interactions in black box models. *CoRR*, abs/1904.00561, 2019.
- G. Cafri and B. A. Bailey. Understanding variable effects from black box prediction: Quantifying effects in tree ensembles using partial dependence. *Journal of Data Science*, 14(1):67–95, 2016.
- D. R. Cutler, T. C. Edwards Jr, K. H. Beard, A. Cutler, K. T. Hess, J. Gibson, and J. J. Lawler. Random forests for classification in ecology. *Ecology*, 88(11):2783–2792, 2007.
- J. Drozdal, J. Weisz, D. Wang, G. Dass, B. Yao, C. Zhao, M. Muller, L. Ju, and H. Su. Trust in AutoML: Exploring information needs for establishing trust in automated machine learning systems. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, pages 297–307, 2020.
- K. Eggenberger, F. Hutter, H. Hoos, and K. Leyton-Brown. Efficient benchmarking of hyperparameter optimizers via surrogates. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, pages 1114–1120, 2015.
- W. D. Fisher. On grouping for maximum homogeneity. *Journal of the American statistical Association*, 53(284):789–798, 1958.
- A. A. Freitas. Automated machine learning for studying the trade-off between predictive accuracy and interpretability. In *Third IFIP International Cross-Domain Conference for Machine Learning and Knowledge Extraction (CD-MAKE 2019)*, volume 11713, pages 48–66. Springer, August 2019.
- J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.
- J. H. Friedman and J. J. Meulman. Multiple additive regression trees with application in epidemiology. *Statistics in medicine*, 22(9):1365–1381, 2003.
- A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, 24(1):44–65, 2015.
- B. M. Greenwell. pdp: An R Package for Constructing Partial Dependence Plots. *The R Journal*, 9(1):421–436, 2017.
- B. M. Greenwell, B. C. Boehmke, and A. J. McCarthy. A simple and effective model-based variable importance measure. *arXiv preprint arXiv:1805.04755*, 2018.
- A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012.
- U. Grömping. Model-agnostic effects plots for interpreting machine learning models. Report 1/2020, Reports in Mathematics, Physics and Chemistry. Department II, Beuth University of Applied Sciences Berlin, 2020.
- F. Hutter, H. H. Hoos, and K. Leyton-Brown. An efficient approach for assessing hyperparameter importance. In *Proceedings of the 31th International Conference on Machine Learning, ICML*, volume 32, pages 754–762. JMLR.org, 2014a.
- F. Hutter, L. Xu, H. H. Hoos, and K. Leyton-Brown. Algorithm runtime prediction: Methods & evaluation. *Artificial Intelligence*, 206:79–111, 2014b.
- J. Levesque, A. Durand, C. Gagné, and R. Sabourin. Bayesian optimization for conditional hyperparameter spaces. In *2017 International Joint Conference on Neural Networks, IJCNN 2017, Anchorage, AK, USA, May 14-19, 2017*, pages 286–293. IEEE, 2017. doi: 10.1109/IJCNN.2017.7965867. URL <https://doi.org/10.1109/IJCNN.2017.7965867>.

-
- C. Molnar, G. König, B. Bischl, and G. Casalicchio. Model-agnostic feature importance and effects with dependent features - A conditional subgroup approach. *CoRR*, abs/2006.04628, 2020.
- R. M. Neil. *Bayesian Learning for Neural Networks*. Springer-Verlag, Berlin, Heidelberg, 1996. ISBN 0387947248.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- F. Pfisterer, J. Thomas, and B. Bischl. Towards human centered AutoML. *CoRR*, abs/1911.02391, 2019.
- P. Probst, A. Boulesteix, and B. Bischl. Tunability: Importance of hyperparameters of machine learning algorithms. *Journal of Machine Learning Research*, 20:53:1–53:32, 2019.
- A. Sharma, J. N. van Rijn, F. Hutter, and A. Müller. Hyperparameter importance for image classification by residual neural networks. In *Discovery Science - 22nd International Conference, DS*, volume 11828 of *Lecture Notes in Computer Science*, pages 112–126. Springer, 2019.
- J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25*, pages 2960–2968, 2012.
- K. Swersky, D. Duvenaud, J. Snoek, F. Hutter, and M. A. Osborne. Raiders of the lost architecture: Kernels for bayesian optimization in conditional parameter spaces. *arXiv: Machine Learning*, 2014.
- J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo. Openml: networked science in machine learning. *SIGKDD Explor.*, 15(2):49–60, 2013.
- M. N. Wright and I. R. König. Splitting on categorical predictors in random forests. *PeerJ*, 7:e6339, 2019.
- I. Xanthopoulos, I. Tsamardinos, V. Christophides, E. Simon, and A. Salinger. Putting the human back in the AutoML loop. In *Proceedings of the Workshops of the EDBT/ICDT 2020 Joint Conference*, volume 2578 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2020.
- L. Zimmer, M. Lindauer, and F. Hutter. Auto-PyTorch Tabular: Multi-fidelity metalearning for efficient and robust AutoDL. *IEEE TPAMI*, 2021. Preprint via Early Access.

Part V.

Conclusion and Open Challenges

8. Conclusion

This thesis addresses the general pitfalls of model-agnostic interpretation methods (Part II). In particular, it investigates two major limitations of global interpretation methods and suggests possible solutions for each of the limitations using grouping (Part III) and partitioning approaches (Part IV), respectively. The main contributions of this thesis can, therefore, be summarized by addressing the subsequent limitations as follows:

1. *Human-incomprehensibility of high-dimensional output.* Most existing interpretation methods are defined for a single feature of interest, which leads in high-dimensional settings to explanations that can be overwhelming and incomprehensible for the user and, therefore, contradictory to achieving transparency, which is the very goal of IML. We addressed this limitation of global feature importance and global feature effect methods by suggesting alternative definitions of these methods based on feature groups in Section 4. We compared different definitions of grouped feature importance methods and provided user guidelines for their practical applicability. Furthermore, we introduced the combined features effects plot, which is based on the concept of PD plots and allows us to visualize the feature effect for a group of features. Thus, the resulting outputs of the interpretation methods are of lower dimension than the outputs of the original feature space, which leads to higher comprehensibility and lower computational costs.

2. *Misleading interpretations of global explanations due to aggregation.* Global interpretation methods are usually the results of an aggregation of underlying local interpretations. Possibly learned feature interactions or extrapolation in sparse regions of the feature space often lead to heterogeneously behaving local interpretations. Thus, calculating the global interpretation method by aggregating over these heterogeneous local interpretations results in an information loss that can cause an aggregation bias. In Section 5, we addressed the aggregation bias due to feature interactions in PD plots for one feature of interest. We introduced a new method based on recursive partitioning, which partitions the feature space into interpretable regions where feature interactions are minimized, and thus, regional PD plots are more representative of the underlying ICE curves. In Section 6, we introduced a general framework that can partition the feature space into interpretable regions such that feature interactions between all (or a subset of) features are minimized. The method can be applied to most feature effect methods, including PD, ALE, and SHAP dependence. In Section 7, we addressed the aggregation bias caused by extrapolation for PD plots in the context of hyperparameter optimization. Therefore, we suggested a recursive partitioning algorithm to obtain regions in the hyperparameter space where resulting regional PD plots for hyperparameter effects can be interpreted more reliably.

9. Open Challenges

9.1. Open Challenges of Grouping Approaches

The suggested grouping approaches provide a lower dimensional output and thus lead to – at least from a dimensionality perspective – more comprehensible results. However, interpretations based on groups of features also lead to an information loss about individual features that cannot be recovered for most methods. While this may be partially possible for the introduced grouped Shapley feature importance method, the computational effort to calculate the individual importance values might be too high for high-dimensional applications. In scenarios that require not only information on the grouped features' influence but also on the individual features' influence, more efficient estimation techniques and implementations to approximate these Shapley feature importance values are needed. Another option might be to provide more selective and human-friendly interpretation outputs on a single feature level, depending on the concrete question the user would like to answer.

The contributing article in Section 4 assumes readily available feature groupings. However, features often cannot naturally be grouped based on domain knowledge, and data-driven grouping approaches might lead to different groupings depending on the chosen approach and might not be meaningful from a domain perspective, which again complicates interpretation. The data-driven approach to group features based on a sparse supervised principle component analysis (Sharifzadeh et al., 2017) showed promising results in our simulation studies since it not only considers dependencies within the feature space but also takes into account the dependencies between the target variable and the features, which might lead to more meaningful groupings for the final interpretation. However, this has yet to be extensively evaluated and compared to other approaches for grouped feature interpretations.

Addressing these open challenges in future work and suggesting a solution that combines the whole interpretation process, from finding meaningful feature groupings to interpretations based on grouped and selective individual features, may be beneficial.

9.2. Open Challenges of Partitioning Approaches

Although the introduced methods in Part IV of the thesis are beneficial for a better understanding of the features' influence on the predictions, they still offer room for improvement. First, it might be difficult for a user to decide on a suitable configuration of the stopping criteria since it is difficult to determine what number of regions is still comprehensive for a user but is also sufficient to obtain representative regional feature effects. Furthermore, it must be considered that the proposed methods are based on the CART algorithm (Breiman et al., 1984), and thus, the deeper we split, the less stable the final tree becomes. One possible solution to obtain more stable results

for these kinds of recursive partitioning algorithms might be to exchange the single decision tree that tends to be unstable with an algorithm that leads to more stable results, such as an ensemble of trees. Another solution could be to limit the choices of potential splits at each node. One possibility to restrict the selection of split features upfront has been presented in the second contributing article of Part IV based on a statistical test for feature interactions.

Hence, providing the user with a more intuitive choice on the configuration of the hyperparameters and improving the algorithm's stability will lead to more interpretable and reliable results and thus might enhance the user's trust. Moreover, the contributing articles in this thesis either focused on minimizing feature interactions or minimizing uncertainty due to extrapolation in sparse regions. An algorithm that considers the simultaneous minimization of feature interactions and uncertainty (or feature correlations) is an interesting challenge for future work.

9.3. General Open Challenges of IML

Although IML is still a young research area, the development of techniques to make ML algorithms more understandable to humans has been impressive in recent years. However, an exponential growth of available IML methods is not necessarily helpful. I will discuss here some general concerns of current research directions in IML.

First, most of the available IML methods are developed from the perspective of ML researchers rather than domain experts. Therefore, whether the method solves a real-world problem or just a theoretical one arises (Du et al., 2019). Adadi and Berrada (2018) also state that there is a difference between explaining and understanding: For understanding, one needs to consider the person receiving the explanation. Hence, when developing IML methods, the ML researcher needs to understand the real-world problem that needs to be solved and how the results can be presented such that the respective explainees understand them. Also, deciding on the granularity of presenting the results to the explainee is a challenging task since there is a trade-off between presenting information in an intuitive way and fully understanding the inner workings of a model (Murdoch et al., 2019). The more we comprise the information (e.g., by looking at PD plots instead of ICE plots), the simpler we can represent it (e.g., by one instead of n curves), but the more information we will lose about the model behavior (e.g., about feature interactions).

Second, evaluating IML methods is difficult since real-world data has no ground-truth interpretation. Thus, many researchers base the evaluation on simple simulation examples for which they know the ground truth. While this evaluation step is essential to judge the faithfulness of an IML method, the evaluation of these methods on real-world data or how well the resulting output of an IML method can be understood by the user (e.g., by user studies) is still lacking in the evaluation process. One reason might be that there is no evaluation approach of IML methods that the community agreed on, as it exists for evaluating ML models. A first step in this direction is, for example, the suggestion of a common taxonomy (Doshi-Velez and Kim, 2017).

I acknowledge that this is not an exhaustive list, but from the perspective of this thesis, the challenges discussed here are important to address in the near future. Therefore, a shared language within the community (Doshi-Velez and Kim, 2017) and more exchange between the different disciplines that develop and apply the methods is necessary. Furthermore, there is a need for more development of systems that allow human-computer interaction to adjust interpretation methods according to the user's understanding and needs (Adadi and Berrada, 2018).

References

- Aas, K., M. Jullum, and A. Løland (2021). Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artificial Intelligence* 298, 103502.
- Adadi, A. and M. Berrada (2018). Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE Access* 6, 52138–52160.
- Angwin, J., J. Larson, S. Mattu, and L. Kirchner (2016). Machine bias. Available at: <https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing> (Accessed: August 25, 2023).
- Apley, D. W. and J. Zhu (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.
- Au, Q., J. Herbringer, C. Stachl, B. Bischl, and G. Casalicchio (2022). Grouped feature importance and combined features effect plot. *Data Mining and Knowledge Discovery* 36(4), 1401–1450.
- Bach, S., A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one* 10(7), e0130140.
- Bianchi, T. (2023). Google - statistics & facts. Available at: <https://www.statista.com/topics/1001/google/#topicOverview> (Accessed: September 8, 2023).
- Blesch, K., D. S. Watson, and M. N. Wright (2023). Conditional feature importance for mixed data. *AStA Advances in Statistical Analysis*, 1–20.
- Bordt, S. and U. von Luxburg (2023). From Shapley values to generalized additive models and back. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, pp. 709–745. PMLR.
- Breiman, L. (2001). Random forests. *Machine Learning* 45(1), 5–32.
- Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone (1984). *Classification and Regression Trees*. Wadsworth.
- Carvalho, D. V., E. M. Pereira, and J. S. Cardoso (2019). Machine learning interpretability: A survey on methods and metrics. *Electronics* 8(8), 832.
- Casalicchio, G., C. Molnar, and B. Bischl (2019). Visualizing the feature importance for black box models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 655–670. Springer.

-
- Catsaros, O. (2023). Generative ai to become a \$1.3 trillion market by 2032, research finds. Available at: <https://www.bloomberg.com/company/press/generative-ai-to-become-a-1-3-trillion-market-by-2032-research-finds/> (Accessed: August 25, 2023).
- Chakraborty, D. and N. R. Pal (2008). Selecting useful groups of features in a connectionist framework. *IEEE Transactions on Neural Networks* 19(3), 381–396.
- Chen, H., J. D. Janizek, S. Lundberg, and S.-I. Lee (2020). True to the model or true to the data? *arXiv preprint arXiv:2006.16234*.
- Covert, I., S. M. Lundberg, and S.-I. Lee (2020). Understanding global feature contributions with additive importance measures. In *Advances in Neural Information Processing Systems*, Volume 33, pp. 17212–17223.
- Dawson, R. J. M. (1995). The “unusual episode” data revisited. *Journal of Statistics Education* 3(3).
- Doshi-Velez, F. and B. Kim (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.
- Du, M., N. Liu, and X. Hu (2019). Techniques for interpretable machine learning. *Communications of the ACM* 63(1), 68–77.
- Fisher, A., C. Rudin, and F. Dominici (2019). All models are wrong, but many are useful: Learning a variable’s importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research* 20(177), 1–81.
- Freiesleben, T., G. König, C. Molnar, and A. Tejero-Cantero (2022). Scientific inference with interpretable machine learning: Analyzing models to learn about real-world phenomena. *arXiv preprint arXiv:2206.05487*.
- Freiesleben, T., C. Molnar, G. König, J. Herbringer, T. Reisinger, G. Casalicchio, M. N. Wright, and B. Bischl (2023). Relating the partial dependence plot and permutation feature importance to the data generating process. *What Does Explainable AI Explain?*.
- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29(5), 1189–1232.
- Friedman, J. H. and B. E. Popescu (2008). Predictive learning via rule ensembles. *The Annals of Applied Statistics* 2(3), 916–954.
- Frye, C., D. de Mijolla, T. Begley, L. Cowton, M. Stanley, and I. Feige (2020). Shapley explainability on the data manifold. *arXiv preprint arXiv:2006.01272*.
- Ghorbani, A. and J. Zou (2019). Data Shapley: Equitable valuation of data for machine learning. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 2242–2251. PMLR.
- Gilpin, L. H., D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal (2018). Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 80–89. IEEE.

- Goldstein, A., A. Kapelner, J. Bleich, and E. Pitkin (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics* 24(1), 44–65.
- Goodman, B. and S. Flaxman (2017). European Union regulations on algorithmic decision-making and a “right to explanation”. *AI Magazine* 38(3), 50–57.
- Grabisch, M. and M. Roubens (1999). An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of Game Theory* 28, 547–565.
- Greenwell, B. M., B. C. Boehmke, and A. J. McCarthy (2018). A simple and effective model-based variable importance measure. *arXiv preprint arXiv:1805.04755*.
- Gregorutti, B., B. Michel, and P. Saint-Pierre (2015). Grouped variable importance with random forests and application to multiple functional data analysis. *Computational Statistics & Data Analysis* 90, 15–35.
- Groombridge, D. (2022). Gartner top 10 strategic technology trends for 2023. Available at: <https://www.gartner.com/en/articles/gartner-top-10-strategic-technology-trends-for-2023> (Accessed: August 25, 2023).
- Hastie, T., R. Tibshirani, J. H. Friedman, and J. H. Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Volume 2. Springer.
- He, Z. and W. Yu (2010). Stable feature selection for biomarker discovery. *Computational Biology and Chemistry* 34(4), 215–225.
- Heaton, J. B., N. G. Polson, and J. H. Witte (2017). Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry* 33(1), 3–12.
- Herbinger, J., B. Bischl, and G. Casalicchio (2022). Repid: Regional effect plots with implicit interaction detection. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, pp. 10209–10233. PMLR.
- Herbinger, J., B. Bischl, and G. Casalicchio (2023). Decomposing global feature effects based on feature interactions. *arXiv preprint arXiv:2306.00541*.
- Herren, A. and P. R. Hahn (2022). Statistical aspects of shap: Functional anova for model interpretation. *arXiv preprint arXiv:2208.09970*.
- Hiabu, M., J. T. Meyer, and M. N. Wright (2023). Unifying local and global model explanations by functional decomposition of low dimensional structures. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, pp. 7040–7060. PMLR.
- Hooker, G. (2004). Discovering additive structure in black box functions. In *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pp. 575–580.
- Hooker, G. (2007). Generalized functional anova diagnostics for high-dimensional functions of dependent variables. *Journal of Computational and Graphical Statistics* 16(3), 709–732.

-
- Janzing, D., L. Minorics, and P. Blöbaum (2020). Feature relevance quantification in explainable ai: A causal problem. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, pp. 2907–2916. PMLR.
- Lei, J., M. G’Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman (2018). Distribution-free predictive inference for regression. *Journal of the American Statistical Association* 113(523), 1094–1111.
- Lengerich, B., S. Tan, C.-H. Chang, G. Hooker, and R. Caruana (2020). Purifying interaction effects with the functional anova: An efficient algorithm for recovering identifiable additive models. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, pp. 2402–2412. PMLR.
- Li, G. and H. Rabitz (2012). General formulation of hdmr component functions with independent and correlated variables. *Journal of Mathematical Chemistry* 50(1), 99–130.
- Lundberg, S. M., G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee (2020). From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence* 2(1), 56–67.
- Lundberg, S. M. and S.-I. Lee (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, Volume 30.
- Mehrabi, N., F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)* 54(6), 1–35.
- Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence* 267, 1–38.
- Molnar, C., G. König, B. Bischl, and G. Casalicchio (2023). Model-agnostic feature importance and effects with dependent features: a conditional subgroup approach. *Data Mining and Knowledge Discovery*, 1–39.
- Molnar, C., G. König, J. Herbringer, T. Freiesleben, S. Dandl, C. A. Scholbeck, G. Casalicchio, M. Grosse-Wentrup, and B. Bischl (2022). General pitfalls of model-agnostic interpretation methods for machine learning models. In A. Holzinger, R. Goebel, R. Fong, T. Moon, K.-R. Müller, and W. Samek (Eds.), *xxAI - Beyond Explainable AI*, Volume 13200 of *Lecture Notes in Artificial Intelligence*, pp. 39–68. Cham: Springer.
- Moosbauer, J., J. Herbringer, G. Casalicchio, M. Lindauer, and B. Bischl (2021). Explaining hyperparameter optimization via partial dependence plots. In *Advances in Neural Information Processing Systems*, Volume 34, pp. 2280–2291.
- Murdoch, W. J., C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu (2019). Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences* 116(44), 22071–22080.
- Obermeyer, Z., B. Powers, C. Vogeli, and S. Mullainathan (2019). Dissecting racial bias in an algorithm used to manage the health of populations. *Science* 366(6464), 447–453.
- OpenAI (2022). Introducing chatgpt. Available at: <https://openai.com/blog/chatgpt> (Accessed: September 6, 2023).

- Peters, M. A. (2018). Deep learning, education and the final stage of automation. *Educational Philosophy and Theory* 50(6-7), 549–553.
- Plachy, O. and T. Vavra (2022). IDC forecasts 18.6% compound annual growth for the artificial intelligence market in 2022-2026. Available at: <https://www.idc.com/getdoc.jsp?containerId=prEUR249536522> (Accessed: August 25, 2023).
- Rahman, S. (2014). A generalized anova dimensional decomposition for dependent probability measures. *SIAM/ASA Journal on Uncertainty Quantification* 2, 670–697.
- Ribeiro, M. T., S. Singh, and C. Guestrin (2016). “Why should I trust you?” Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144.
- Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1(5), 206–215.
- Scholbeck, C. A., C. Molnar, C. Heumann, B. Bischl, and G. Casalicchio (2020). Sampling, intervention, prediction, aggregation: a generalized framework for model-agnostic interpretations. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 205–216. Springer.
- Schwalbe, G. and B. Finzel (2023). A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts. *Data Mining and Knowledge Discovery*, 1–59.
- Selvaraju, R. R., M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 618–626.
- Shapley, L. S. (1953). A value for n-person games. *Contributions to the Theory of Games* 2(28), 307–317.
- Sharifzadeh, S., A. Ghodsi, L. H. Clemmensen, and B. K. Ersbøll (2017). Sparse supervised principal component analysis (sspca) for dimension reduction and variable selection. *Engineering Applications of Artificial Intelligence* 65, 168–177.
- Stoll, J. (2023). Number of Netflix paid subscribers worldwide from 1st quarter 2013 to 2nd quarter 2023. Available at: <https://www.statista.com/statistics/250934/quarterly-number-of-netflix-streaming-subscribers-worldwide/> (Accessed: September 8, 2023).
- Stone, C. J. (1994). The use of polynomial splines and their tensor products in multivariate function estimation. *The Annals of Statistics* 22(1), 118 – 171.
- Štrumbelj, E. and I. Kononenko (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems* 41, 647–665.
- Sun, X., Z. Wang, R. Ding, S. Han, and D. Zhang (2022). Puregam: Learning an inherently pure additive model. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1728–1738.

- Sundararajan, M. and A. Najmi (2020). The many Shapley values for model explanation. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 9269–9278.
- Topol, E. J. (2019). High-performance medicine: the convergence of human and artificial intelligence. *Nature medicine* 25(1), 44–56.
- Tsai, C.-P., C.-K. Yeh, and P. Ravikumar (2023). Faith-shap: The faithful Shapley interaction index. *Journal of Machine Learning Research* 24(94), 1–42.
- Watson, D. S. (2022). Conceptual challenges for interpretable machine learning. *Synthese* 200(2), 65.
- Watson, D. S. and M. N. Wright (2021). Testing conditional independence in supervised learning algorithms. *Machine Learning* 110(8), 2107–2129.
- Wong, B. (2023). Top social media statistics and trends of 2023. Available at: <https://www.forbes.com/advisor/business/social-media-statistics/#source> (Accessed: September 8, 2023).

Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12. Juli 2011, §8 Abs. 2 Pkt. 5)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

München, den 22.09.23

Julia Herbinger

