

How to safely discard features based on aggregate SHAP values

Robi Bhattacharjee*

ROBI.BHATTACHARJEE@WSII.UNI-TUEBINGEN.DE

University of Tübingen and Tübingen AI Center

Karolin Frohnappfel*

KAROLIN.FROHNAPFEL@UNI-TUEBINGEN.DE

University of Tübingen

Ulrike von Luxburg

ULRIKE.LUXBURG@UNI-TUEBINGEN.DE

University of Tübingen and Tübingen AI Center

Abstract

SHAP is one of the most popular *local* feature-attribution methods. Given a function f and an input $x \in \mathbb{R}^d$, it quantifies each feature’s contribution to $f(x)$. Recently, SHAP has been increasingly used for *global* insights: practitioners average the absolute SHAP values over many data points to compute global feature importance scores, which are then used to discard “unimportant” features. In this work, we investigate the soundness of this practice by asking whether small aggregate SHAP values necessarily imply that the corresponding feature does not affect the function. Unfortunately, the answer is no: even if the i -th SHAP value equals 0 on the entire data support, there exist functions that clearly depend on Feature i . The issue is that computing SHAP values involves evaluating f on points outside of the data support, where f can be strategically designed to mask its dependence on Feature i . To address this, we propose to aggregate SHAP values over the *extended* support, which is the product of the marginals of the underlying distribution. With this modification, we show that a small aggregate SHAP value implies that we can safely discard the corresponding feature. We then extend our results to KernelSHAP, the most popular method to approximate SHAP values in practice. We show that if KernelSHAP is computed over the extended distribution, a small aggregate KernelSHAP value justifies feature removal. This result holds independently of whether KernelSHAP accurately approximates true SHAP values, making it one of the first theoretical results to characterize the KernelSHAP algorithm itself. Our findings have both theoretical and practical implications. We introduce the “Shapley Lie algebra”, which offers algebraic insights that may enable a deeper investigation of SHAP and we show that a simple pre-processing step – randomly permuting each column of the data matrix – enables safely discarding features based on aggregate SHAP and KernelSHAP values.

Keywords: Interpretability, Explainable machine learning, XAI, Shapley values, feature selection, Lie Theory

1. Introduction

Due to the widespread adoption of large, opaque models, explainability has become an essential topic in machine learning. One particularly prominent application domain is *scientific discovery*, where practitioners train a model not only for accurate predictions but also to gain insight into their specific problem and its underlying mechanisms. In such cases, the true value of the machine learning model lies in the understanding it provides, making interpretability techniques critical.

In science, SHAP (Lundberg and Lee, 2017) is by far the most widely used method for generating explanations. It is a local feature-attribution method that is applied across various fields

. *Equal contribution

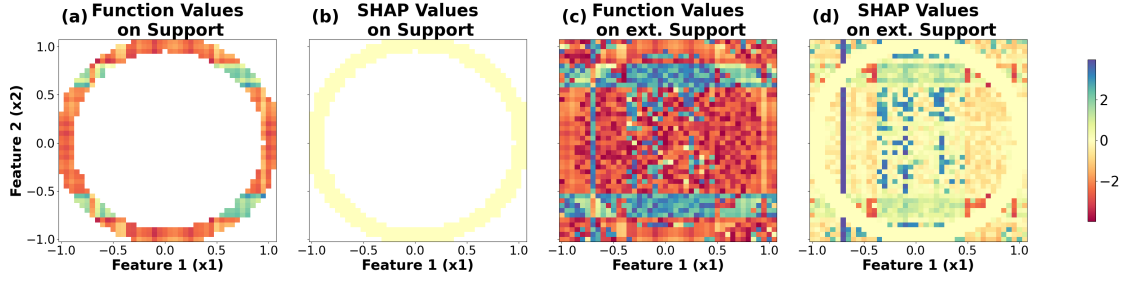


Figure 1: Example of a function where the aggregate SHAP value of Feature 1 is 0, yet the function depends on this feature. (a): Function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, supported on a ring with the color depicting the function value. The function clearly depends on both Features 1 and 2. **(b):** Point-wise SHAP values $\phi_1(\mu, f, x)$ of Feature 1 are constantly 0 on the support. This provides the counter-example we have been looking for. **(c) and (d):** Function and SHAP values on the extended support. Here the SHAP values are not constantly 0 any more, illustrating the direction towards resolving the issue of the counter-example.

including biology (Berdugo et al., 2022), geoscience (Jiang et al., 2024), medicine (Martínez-Ruiz et al., 2023), psychiatry (Giuntella et al., 2021), physics (Li et al., 2021), and chemistry (Wojtuch et al., 2021). SHAP operates as follows: for an input distribution μ over \mathbb{R}^d , a model $f : \mathbb{R}^d \rightarrow \mathbb{R}$, and a fixed input point $x \in \mathbb{R}^d$, it outputs d values, $\phi_1(\mu, f, x), \dots, \phi_d(\mu, f, x)$ that quantify the “impact” that each feature had in predicting $f(x)$. There are a variety of ways these values can be defined and implemented. In this work, we exclusively focus on *interventional SHAP*, which is the more widely used definition, and on *KernelSHAP* (Lundberg and Lee, 2017), which is the most popular algorithm for approximating it.

When it was invented, SHAP was clearly designed towards *local* explanations, which apply to a specific input point. However, especially in scientific contexts, SHAP has recently become popular for providing *global* feature importance, for example in Greenwood et al. (2024), Sharma Timilsina et al. (2024), Bernard et al. (2023), Delavaux et al. (2023), Chen et al. (2022), Ekanayake et al. (2022), Qiu et al. (2022), Rane et al. (2022), Wang et al. (2022) or Yang et al. (2022), see also Appendix G. Practitioners average the *absolute* SHAP values $|\phi_i(\mu, f, x)|$ over many data points drawn from the input distribution μ to obtain d *aggregate SHAP values* $\overline{\phi}_1(\mu, f), \dots, \overline{\phi}_d(\mu, f)$. Features are then typically sorted, selected, and globally interpreted based on these aggregate SHAP values. Despite its popularity, this method lacks any theoretical guarantees. In this work, we address this gap by studying a fundamental property we call *soundness*, which means that features with a low global importance score are not impactful on making predictions. More specifically, we seek to answer the following question:

If $\overline{\phi}_i(\mu, f)$ is small, does that mean Feature i is globally irrelevant for predictions made by f ?

1.1. Our Contributions

In Section 3.1 we begin with the extreme version of this problem where $\overline{\phi}_i(\mu, f)$ precisely equals 0. Our question then becomes the following: if $\overline{\phi}_i(\mu, f) = 0$, can f be computed without *any* access to x_i ? Unfortunately, the answer to this question is no. As shown in Panels (a) and (b) of Figure 1, there exist examples where the SHAP value $\phi_i(\mu, f, x)$ is constantly 0 across $\text{supp}(\mu)$ (Panel (b)) and yet f clearly exhibits variation across x_i within $\text{supp}(\mu)$ (Panel (a)). The core issue is that computing

Algorithm 1 Sound Aggregate KernelSHAP

Input: $X \in \mathbb{R}^{n \times d}$: data matrix; f : function; $i \in [d]$: feature of interest

- 1: **for** $j \in [d]$ **do**
 - 2: Randomly permute j -th column: $\tilde{X}_j \leftarrow \text{Permute}(X_j)$
 - 3: **end for**
 - 4: Save shuffled data matrix: $\tilde{X} \leftarrow (\tilde{X}_1, \dots, \tilde{X}_d)$
 - 5: Calculate local SHAP values: $\phi_i^{(1)}, \dots, \phi_i^{(n)} \leftarrow \text{KernelSHAP}(\tilde{X}, f)$
 - 6: Aggregate: $\bar{\phi}_i \leftarrow \frac{1}{n} \sum_{k=1}^n |\phi_i^{(k)}|$
-

$\phi_i(\mu, f, x)$ requires evaluating f on points that are potentially *outside* $\text{supp}(\mu)$, and those values can be strategically chosen to cause $\phi_i(\mu, f, x)$ to equal 0 *inside* $\text{supp}(\mu)$ (see Figure 1).

In Section 3.2, we expand on this observation and introduce the notion of the *extended support* $\text{supp}(\mu^*)$, which is the product of the supports of the marginal distributions μ_i of μ across each feature. Then, in our first result (Theorem 6), we show that constant-zero aggregate SHAP values over the extended support $\text{supp}(\mu^*)$ is sufficient for discarding a feature. This is illustrated in Panel (d) of Figure 1 where the dependence of f on x_1 becomes apparent when looking at SHAP values over $\text{supp}(\mu^*)$.

In Section 3.4, we extend our result from the extreme case where $\bar{\phi}_i(\mu, f) = 0$ to cases where this approximately holds. To incorporate the entire extended support, we propose aggregating and computing SHAP values fully using the extended distribution μ^* . We then show (Theorem 11) that doing so gives a more robust bound on the impact of removing a feature with a small aggregate SHAP value $\bar{\phi}_i(\mu^*, f)$.

In Section 4 we turn our attention to the finite sample regime where SHAP values are computed with respect to $X \sim \mu^n$. This setting poses an additional challenge: there are no known results bounding how well the most popular algorithm for computing SHAP values, KernelSHAP, approximates the true SHAP values. Surprisingly, our techniques completely circumvent this by proving the first known soundness result *directly* about KernelSHAP. In Theorem 12, we show that when KernelSHAP values are computed over a data sample from the extended distribution μ^* , a small aggregate value implies that the feature has a small impact on the prediction. Furthermore, we observe that sampling from μ^* can be easily implemented in practice – simply permute each feature column of a data matrix that is sampled from the original distribution μ (lines 1-3 of Algorithm 1). With this simple modification, our theorem *directly applies* to real-life implementations of KernelSHAP.

We believe our work has interesting implications both in theory and in practice. From the theoretical side, our main contributions are:

- Two theorems (Theorems 6 and 11) characterizing when we can safely discard features using aggregate SHAP values.
- The first soundness analysis of KernelSHAP that holds *independently* of how well KernelSHAP approximates the true SHAP values.
- A novel technical tool we call the Shapley Lie algebra (Definition 14). This construction captures many useful algebraic properties of SHAP values which are central to proving our main results. We believe our techniques might be useful to studying other properties of SHAP and KernelSHAP as well.

We also note that our work is *not* intended to exclusively provide novel algorithms for feature selection: there exist other approaches for doing so. Instead, *this work characterizes the soundness of an approach that is already widely used in practice*. Our idea of using the extended distribution μ^* is intended to provide a theoretically justified modification of SHAP that enjoys provable soundness while (hopefully) preserving other desirable aspects of the algorithm.

For practitioners, our work has a very clear and simple implication: to discard features based on aggregate SHAP values, it is not enough to average SHAP values over a (subset of) the original data points. Instead, one has to sample from the extended support, which luckily is very simple by randomizing features, as can be seen in the pseudo-code above (Algorithm 1).

1.2. Related Work

SHAP values: Since their introduction to the machine learning community by [Lundberg and Lee \(2017\)](#) SHAP values, which originate from game theory ([Shapley, 1953](#)), have gained increasing attention. See for example [Lundberg et al. \(2020\)](#), [Covert et al. \(2020\)](#), [Frye et al. \(2020\)](#) and [Bordt and von Luxburg \(2023\)](#), just to state a few. Additionally to SHAP values we also look into KernelSHAP ([Lundberg and Lee, 2017](#), [Covert and Lee, 2021](#)), which is the most widely used approximation algorithm for SHAP values. Similar to us, [Slack et al. \(2020\)](#) exploit the fact that interventional SHAP values are calculated using data points outside the distribution, however, with the different goal of masking an unfair algorithm as fair. Also [Merrick and Taly \(2020\)](#) and [Kumar et al. \(2020\)](#) consider this when investigating the axioms of SHAP values.

Global Feature Importance: In explainable machine learning many global feature importance methods exist, such as LOCO ([Lei et al., 2018](#)) or SAGE ([Covert et al., 2020](#)). However, in practice scientists also tend to simply aggregate local feature attributions to get global insights. While the focus of this paper is on the aggregation of SHAP values, aggregating other feature attribution methods such as LIME ([van der Linden et al., 2019](#)) and Anchors ([Mor et al., 2024](#)) has been proposed as well. The idea of using explainability techniques for feature importance has become a subject of ongoing research ([Hooker et al., 2019](#), [Merrick and Taly, 2020](#), [Kumar et al., 2020](#), [Ewald et al., 2024](#), [Verdinelli and Wasserman, 2024](#)) and some also investigate the possibility of performing feature selection ([Marcilio and Eler, 2020](#)) or data selection ([Wang et al., 2024](#)).

Explainable Machine Learning: While computing SHAP values is one of the most widely used methods of feature attribution in explainable machine learning (see [Molnar \(2022\)](#) for an overview), many others exist, such as LIME ([Ribeiro et al., 2016](#)), Integrated Gradients ([Sundararajan et al., 2017](#)) and Anchors ([Ribeiro et al., 2018](#)). The literature on these methods is vast with a lot of work dedicated on giving theoretical guarantees for feature attribution methods. See for example [Dasgupta et al. \(2022\)](#), [Bilodeau et al. \(2024\)](#) and [Bressan et al. \(2024\)](#).

2. Preliminaries

2.1. Notation

We consider explanations for functions $f : \mathbb{R}^d \rightarrow \mathbb{R}$. For $x \in \mathbb{R}^d$, we let (x_1, \dots, x_d) denote its coordinates. For a subset of indices $S \subseteq [d] = \{1, \dots, d\}$, we let S^c denote its complement and \mathbb{R}^S denote the projection of \mathbb{R}^d onto its coordinates in S . That is, for $x \in \mathbb{R}^d$ we let $x_S = (x_i : i \in S) \in \mathbb{R}^S$. It will be useful to apply functions whose coordinates are drawn from different

points. To denote this, if $x^{(1)}, \dots, x^{(k)} \in \mathbb{R}^d$ are k points and $S^{(1)}, \dots, S^{(k)}$ are k disjoint subsets that partition $[d]$, then we let $f\left(x_{S^{(1)}}^{(1)}, \dots, x_{S^{(k)}}^{(k)}\right) = f(x)$ where x is the unique point such that $x_{S^{(i)}} = x_{S^{(i)}}^{(i)}$.

2.2. The SHAP explanation method

SHAP is a local posthoc explanation method that generates a separate explanation for each individual prediction. Given a data distribution μ , a function f , and a data point x , it calculates d SHAP values which quantify the contribution of each feature to the output of f at x . To do so, it makes use of a *value function* $v_S(\mu, f, x)$ that associates each subset $S \subseteq [d]$ of features with the prediction $f(x)$. v_S is intended to simulate the behavior that f might have if it only had access to features inside S . The current literature typically considers two main choices of value functions.

Definition 1 (Value Function) *Let μ be a distribution over \mathbb{R}^d , $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be a function, and $x \in \mathbb{R}^d$ a point. Let $S \subseteq [d]$ be a subset of features. The observational and interventional value functions corresponding to S are defined as*

$$\begin{aligned} v_S^{obs}(\mu, f, x) &= \mathbb{E}_{X \sim \mu}[f(X) | X_S = x_S], \\ v_S^{int}(\mu, f, x) &= \mathbb{E}_{X \sim \mu}[f(x_S, X_{S^c})]. \end{aligned}$$

v_S^{obs} represents the value of f when feature values outside of S are sampled from the conditional distribution, while v_S^{int} does the same using the marginal distribution. Due to the difficulty of sampling from a conditional distribution, the interventional value function is more widely used in practice, and from this point forward we will exclusively use it. To simplify notation, we will simply write v_S to mean v_S^{int} .

Definition 2 (SHAP Values) *Let μ be a distribution over \mathbb{R}^d , $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be a function, and $x \in \mathbb{R}^d$ a point. For $1 \leq i \leq d$, the i th SHAP value of f at x is defined as*

$$\phi_i(\mu, f, x) = \frac{1}{d} \sum_{S \subseteq [d] \setminus \{i\}} \binom{d-1}{|S|}^{-1} (v_{S \cup \{i\}}(\mu, f, x) - v_S(\mu, f, x)).$$

SHAP values are designed to effectively distill the information provided by the value function over all 2^d possible subsets of features into one value ϕ_i per feature.

Although SHAP values are primarily a local explanation method, they are increasingly used to derive global, feature-based explanations for machine learning models. This is typically achieved by averaging the absolute values of SHAP values for a given feature across the entire data distribution.

Definition 3 (Aggregate SHAP Values) *Let μ, f be a distribution and a function. Then the aggregate SHAP values $\bar{\phi}_i(\mu, f)$ are defined as $\bar{\phi}_i(\mu, f) = \mathbb{E}_{x \sim \mu} |\phi_i(\mu, f, x)|$.*

Practitioners typically interpret these values by discarding features with small aggregate SHAP values and concentrating on those with relatively large ones. The main purpose of this paper is to investigate how sound this practice is. We now formalize what it means to be able to safely “discard” a feature.

Definition 4 (Determined Function / Discarding Features) *Let $S \subseteq [d]$ be a set of indices, $f : \mathbb{R}^d \rightarrow \mathbb{R}$ a function, and $\mathcal{X} \subseteq \mathbb{R}^d$ a subset. f is S -determined over \mathcal{X} if for all $a, b \in \mathcal{X}$, $a_S = b_S \implies f(a) = f(b)$. We say that Feature i can be discarded for function f over \mathcal{X} if f is*

$[d] \setminus \{i\}$ -determined over \mathcal{X} . Additionally, we set the convention that a \emptyset -determined function is a constant function.

Intuitively, we can discard a feature if it does not “influence” the outcome of the function on the data support.

3. Characterization of Aggregate SHAP Values

The main question of this paper is to investigate whether features that have small aggregate SHAP values can be safely discarded or not. Surprisingly, and opposed to current practice in data science, the answer to this question is no. Let us show a simple counter-example.

3.1. Constant-zero SHAP values on the entire support do not allow to discard features

We consider a data-generating distribution μ with support on a two-dimensional ring, and a function f that is defined on this support (Figure 1, Panel (a)). The distribution μ and the function f are chosen in such a way that the pointwise SHAP values $\phi_1(\mu, f, x)$ of Feature 1 are constantly 0 on the support of μ , so in particular the aggregate SHAP value $\phi_1(\mu, f)$ is 0. Yet, the function f obviously is not independent of Feature 1, hence this feature cannot be discarded. The key to achieving this behavior is the fact that we use interventional SHAP: to compute the value functions, we sample from the marginal distributions of both features, whose supports extend beyond the ring. In our example, this allowed us to strategically choose the function values of f on this “extended support” such that the SHAP values *within the support* are constantly 0. Observe that the SHAP values in these out-of-support regions are no longer 0 (Figure 1, Panel (d)). To construct this example, we used a linear program. See Appendix F for details and more examples with a similar behavior.

3.2. Constant-zero SHAP values over the extended support allow to discard features

Contemplating the counter-example above leads to the following idea: to understand whether we can discard Feature i , we need to look at its SHAP values *beyond* the support of the data distribution. To this end, we begin by defining a natural distribution associated with μ that characterizes precisely where we look. This “extended distribution” is constructed to have each feature independently range over its entire support according to μ . It is formally defined as follows:

Definition 5 (Extended distribution and extended support) *Let μ be a distribution over \mathbb{R}^d , and let μ_i denote its marginal distribution of Feature i . Let $\mu_1^*, \mu_2^*, \dots, \mu_d^*$ denote independent distributions such that μ_i^* is identically distributed to μ_i . Then the extended distribution μ^* is defined as the product distribution $\mu^* = \prod_{i=1}^d \mu_i^*$. We call the support of the extended distribution $\text{supp}(\mu^*)$ the extended support.*

The obvious question is now whether a constant-zero SHAP value across the *extended* support implies that a feature can be safely discarded? We answer this affirmatively in the following theorem, which is the first main result of this paper.

Theorem 6 (Discarding features based on constant-zero SHAP values on extended support)

Let μ be a distribution on \mathbb{R}^d and $f : \mathbb{R}^d \rightarrow \mathbb{R}$ a measurable function. Let $1 \leq i \leq d$ be a feature. Then f is $[d] \setminus \{i\}$ -determined over $\text{supp}(\mu^)$ if and only if $\phi_i(\mu, f, x) = 0$ for all $x \in \text{supp}(\mu^*)$.*

Note that this theorem concerns SHAP values $\phi_i(\mu, f, x)$ with respect to the original distribution μ , but we need to consider these values for all points x in the extended support $\text{supp}(\mu^*)$. More

generally, the theorem would equally hold if we considered the SHAP values $\phi_i(\mu^*, f, x)$ instead. Additionally, this theorem implies that when μ has full support on \mathbb{R}^d , a constant-zero SHAP value is a sufficient condition for discarding a feature.

3.3. Proof of Theorem 6

Let F denote the vector space of all measurable functions from $\text{supp}(\mu^*) \rightarrow \mathbb{C}$ (we generalize to functions ranging over complex values for technical reasons based on the nicer properties of complex vector spaces). To prove Theorem 6, we begin by reframing it as a statement about linear operators that act on F . To do so, we use the following definitions.

Definition 7 (Determined Function Space) *Let F_S denote the vector space of all measurable S -determined functions from $\text{supp}(\mu^*) \rightarrow \mathbb{C}$.*

Observe that F_S is a vector space because linear combinations of S -determined functions are S -determined themselves. Next, we show (proof in Appendix A.2) that there exist linear operators $F \rightarrow F$ that correspond to value functions (Definition 1) and SHAP values (Definition 2).

Lemma 8 (Value operator) *Let $S \subseteq [d]$. There exists a linear operator $v_S : F \rightarrow F$ such that $(v_S f)(x) = v_S(\mu, f, x)$ for all $f \in F$.*

Definition 9 (SHAP operator) *Let $1 \leq i \leq d$ be a feature. Then the SHAP operator $\Phi_i : F \rightarrow F$ is defined as $\Phi_i f = A_i f - B_i f$ where*

$$A_i f = \frac{1}{d} \sum_{S \subseteq [d] \setminus \{i\}} \binom{d-1}{|S|}^{-1} v_{S \cup \{i\}} f \text{ and } B_i f = \frac{1}{d} \sum_{S \subseteq [d] \setminus \{i\}} \binom{d-1}{|S|}^{-1} v_S f.$$

Expanding out this definition immediately implies that $(\Phi_i f)(x)$ is precisely the i th SHAP value $\phi_i(\mu, f, x)$ (Definition 2) of f at x with respect to μ . Thus we can reframe the statement of Theorem 6 as follows: for any $f \in F$, $\Phi_i f = 0 \iff f \in F_{[d] \setminus \{i\}}$. Our strategy to prove this will be to derive a useful set of properties of value operators that culminate in the following characterizations of A_i and B_i .

Lemma 10 (Properties of A_i and B_i) *The operators A_i and B_i satisfy the following properties:*

1. **Image of A_i :** For all $S \subseteq [d]$, $A_i(F_S) \subseteq F_S$.
2. **Image of B_i :** $B_i(F) \subseteq F_{[d] \setminus \{i\}}$.
3. **Kernel of A_i :** $A_i^{-1}(\{0\}) = \{0\}$.

Properties 1 and 2 demonstrate that A_i and B_i both tend to preserve determined functions, while Property 3 implies that A_i has a trivial kernel. Lemma 10 is a consequence of the algebraic structure of the value operators, and its proof is surprisingly involved. As we will see, the set of value operators $\{v_S : S \subseteq [d]\}$ forms a solvable Lie algebra, which provides useful structure to prove the lemma. We defer the proof of Lemma 10 to Section 5. Instead, let us show how this lemma implies Theorem 6.

Proof [Theorem 6 (Sketch); full proof in Appendix A.1] According to our previous discussion, it suffices to show that for all $f \in F$ and $1 \leq i \leq d$, $\Phi_i f = 0$ if and only if $f \in F_{[d] \setminus \{i\}}$. The “ \Leftarrow ” direction is straightforward (see appendix), so here we focus our attention on the “ \Rightarrow ” direction. Suppose $\Phi_i f = 0$, which means $A_i f = B_i f$. Property 2 of Lemma 10 implies $B_i f \in F_{[d] \setminus \{i\}}$, and

thus $A_i f \in F_{[d] \setminus \{i\}}$. Thus, it suffices to show that the pre-image of $F_{[d] \setminus \{i\}}$ (denoted $A_i^{-1}(F_{[d] \setminus \{i\}})$) is a subset of $F_{[d] \setminus \{i\}}$.

Doing so is particularly simple when F (and therefore $F_{[d] \setminus \{i\}}$) is finite dimensional. Property 3 of Lemma 10 implies that A_i has a trivial kernel, which means that A_i has a well defined inverse A_i^{-1} . Property 1 of Lemma 10 implies that $A_i(F_{[d] \setminus \{i\}}) \subseteq F_{[d] \setminus \{i\}}$. In the case where F is finite dimensional, it then follows that $\dim(A_i(F_{[d] \setminus \{i\}})) = \dim(F_{[d] \setminus \{i\}})$, which implies that the two vector spaces must be equal. Thus A_i is an injective and surjective map from $F_{[d] \setminus \{i\}}$ to itself, which means that it must be bijective, which implies $A_i^{-1}(F_{[d] \setminus \{i\}}) \subseteq F_{[d] \setminus \{i\}}$. To handle the infinite-dimensional case, it turns out there is a technical trick one can use to reduce it to the finite dimensional case. We defer this to Appendix A.1. ■

3.4. Discarding features based on close-to-zero aggregate SHAP values

Theorem 6 has two drawbacks. First, it requires SHAP values to be *exactly* equal to 0. Second, it considers all points in the entire extended support. Thus translating it into a statement about aggregate SHAP values is not immediately obvious, because aggregate SHAP values are averaged over the support of the original distribution. We address both of these issues by replacing μ with the extended distribution μ^* . That is, we propose that aggregate SHAP values be computed with respect to the extended distribution. This immediately addresses the second issue as these aggregate values *will* take the full extended support into account. It turns out, this idea also addresses the first issue, allowing for a more flexible bound.

Theorem 11 (Small μ^* -SHAP value allows to discard feature) *Let μ be a distribution on \mathbb{R}^d , and $f : \mathbb{R}^d \rightarrow [0, 1]$ a measurable function. Let $1 \leq i \leq d$ be a feature. Suppose that the aggregate SHAP value, $\overline{\phi}_i(\mu^*, f) \leq \epsilon$. Then there exists $g \in F_{[d] \setminus \{i\}}$ s.t. $\int (f(x) - g(x))^2 d\mu^*(x) < d^2 \epsilon$.*

Observe here that the SHAP values are both averaged over and computed with μ^* . In addition to encompassing the entire extended support, we will see that μ^* also lends itself to a tighter analysis due to its features being independent.

Proof [Theorem 11 (Sketch); full proof in Appendix A.4] Recall that F denotes the space of all measurable functions $\text{supp}(\mu^*) \rightarrow \mathbb{C}$. The key observation is to define an inner product over F with $\langle f_1, f_2 \rangle = \int f_1(x) \overline{f_2(x)} d\mu^*(x)$. We can then show that over μ^* , the value operators v_S are *Hermitian*. From here, we can essentially follow the proof of Theorem 6. The only difference is that when we apply Lemma 10, we can additionally bound the eigenvalues of A_i^{-1} (thus strengthening Property 3 of Lemma 10). We then conclude by arguing that if $(A_i - B_i)f$ is close to 0, then $A_i f$ is close to $F_{[d] \setminus \{i\}}$. This means that the distance from f to $F_{[d] \setminus \{i\}}$ can be bounded with the norm of A_i^{-1} , which in turn is bounded based on its eigenvalues (as it too is Hermitian). ■

4. Aggregate SHAP Values in the Finite Sample Setting

Thus far, our results have been in the distributional setting where SHAP values and value functions are both computed based on the true expectations taken over μ (or μ^*). Hence, as a next step we will study the *finite sample regime*, where SHAP values are computed with respect to an i.i.d sample $X = \{x^{(1)}, \dots, x^{(n)}\} \sim \mu^n$. A natural way to approximate SHAP values in this setting is to replace

true expectations with their corresponding empirical estimates. However, this is computationally infeasible in practice due to the exponential number of subsets S (2^d total) one must consider. The most popular method to address this issue is KernelSHAP (Lundberg and Lee, 2017), which uses weighted linear regression to approximate the value function $v_S(\mu, f, x)$, and then combines these approximations to obtain a tractable estimate of the SHAP value $\phi_i(\mu, f, x)$. For the purposes of proving our results, we include a detailed definition of KernelSHAP in Appendix B.1.

We denote the KernelSHAP value for Feature i at point $x \in \mathbb{R}^d$ with $\mathcal{K}_i(X, f, x)$, where $X = \{x^{(1)}, \dots, x^{(n)}\}$ is a set of n points in \mathbb{R}^d . We also denote the aggregate KernelSHAP value as $\bar{\mathcal{K}}_i(X, f)$ which is defined as

$$\bar{\mathcal{K}}_i \left(X = \{x^{(1)}, \dots, x^{(n)}\}, f \right) = \frac{1}{n} \sum_{j=1}^n |\mathcal{K}_i(X, f, x^{(j)})|.$$

We now turn to our main objective, which is to find an analog of Theorem 11 that applies to KernelSHAP. Recall that the main idea from the previous section was that SHAP values must be computed over the *extended distribution* μ^* in order to achieve soundness. This idea will also apply to KernelSHAP in a similar way. To use the extended distribution, we need to replace the training sample $X \sim \mu^n$ with a sample from $(\mu^*)^n$. Although this cannot be directly done (as typically users only have access to samples from μ), it turns out that simply scrambling the columns of the data matrix X (as shown in Algorithm 1) suffices. More precisely, we let $X^* = \{(x^*)^{(1)}, \dots, (x^*)^{(n)}\}$ be the dataset constructed as follows: if $\sigma_1, \dots, \sigma_d$ are independent random permutations of $[n] = \{1, \dots, n\}$, then

$$(x^*)_i^{(j)} = x_i^{(\sigma_i(j))} : 1 \leq i \leq d, 1 \leq j \leq n. \quad (1)$$

We now investigate the soundness of running KernelSHAP over this scrambled dataset. To do so, we will express our results in terms of an error term, denoted $\eta(X^*, \mu^*, f)$. This term represents how far the aggregate SHAP values are from the values that *KernelSHAP* converges towards in the large sample limit. Crucially, this term has no relevance to the true SHAP values – it is rather a reflection of the convergence behavior that KernelSHAP exhibits when it is applied over a large dataset. This quantity is extensively studied in (Covert and Lee, 2021), and has been shown to be quite small both theoretically and practically. We include a full discussion of this in Appendix B.1.

Theorem 12 (Small μ^* -aggregate KernelSHAP Value allows to discard features) *Let μ be a distribution and $f : \mathbb{R}^d \rightarrow [0, 1]$ a measurable function. Let $1 \leq i \leq d$ be a feature. Let $X = \{x^{(1)}, \dots, x^{(n)}\} \sim \mu^n$ denote an i.i.d. sample of n points from μ , and let X^* be as defined in Equation 1. Suppose that $\bar{\mathcal{K}}_i(X^*, f) \leq \epsilon$. Then there exists $g \in F_{[d] \setminus \{i\}}$ such that*

$$\int (f(x) - g(x))^2 d\mu^*(x) < d^2 (\epsilon + \eta(X^*, \mu^*, f)),$$

where $\eta(X^*, \mu^*, f)$ denotes the error between the empirical computation of KernelSHAP and its limit object (see Definition 25).

Theorem 12 has *direct* implications for practitioners: implementing a procedure for constructing X^* is trivial and this is the *only* modification needed for KernelSHAP to enjoy similar soundness guarantees as SHAP does. We now briefly sketch a proof, with full details deferred to Appendix B.2.

Proof [Theorem 12 (Sketch); full proof in Appendix B.2] We define an operator, \mathcal{K}_i , that corresponds to the limit object of KernelSHAP. The crux of this proof is to use the explicit formula for \mathcal{K}_i given in Covert and Lee (2021) to show that much like SHAP operator Φ_i , \mathcal{K}_i can be expressed as a linear combination of value operators. This allows us to leverage an analog of Lemma 10. At a high level, the limit object of KernelSHAP is the solution to a particular linear regression that attempts to predict value functions. Using the standard formula for solving a linear regression, we see that this solution is *linear* with respect to the target vector. This implies that \mathcal{K}_i itself is linear with respect to the value functions. Finally, we show that this linear combination satisfies an analog of Lemma 10. This follows from straightforward algebraic manipulations, beginning with an explicit formula for $\mathcal{K}_i f$ derived in Covert and Lee (2021). ■

5. Technical Toolbox: The Shapley Lie Algebra

We now develop the technical tools that allow us to prove Lemma 10, the key ingredient in our proofs. In Section 5.1, we prove Properties 1 and 2 by studying the relationship between value operators and determined spaces. Observe that the operators A_i and B_i (Definition 9) are linear combinations of value operators, and thus their behavior over determined spaces can be characterized based on looking at value operators.

In Section 5.2, we develop the technical machinery for proving Property 3. The core difficulty is to analyze the invertibility of a linear combination of linear operators. One natural idea for doing so would be to attempt to simultaneously diagonalize the operators. This would allow us to only consider diagonal matrices, which would greatly simplify the problem. Unfortunately simultaneous diagonalization is only possible for a set of commuting matrices. To circumvent this, we will appeal to Lie Theory which provides tools to study families of transformations that *almost commute* with each other. First, we construct a Lie algebra generated by the value operators and show that it is *solvable* (which can be thought of as a generalization of commutative). Second, we apply Lie’s theorem to find a basis in which all value operators are simultaneously *upper triangular*. This enables us to show that A_i is also upper triangular, which in turn demonstrates its invertibility.

5.1. Properties of Value Operators

Recall that the idea behind the value function $v_S(\mu, f, x)$ is to represent what f would output at x if it only had access to the coordinates from x_S . Applying this over all $x \in \text{supp}(\mu^*)$ suggests that applying the value operator v_S to f results in a function $v_S f$ that is only impacted by the coordinates of its input from S . Phrasing this in terms of determined function spaces, this can be written as $v_S f \in F_S$. We now study the more general problem of characterizing how v_S behaves over an arbitrary determined function space F_T for some other set $T \subseteq [d]$.

Lemma 13 (Images and Eigenspaces of Value Operators) *If $S, T \subseteq [d]$, then the following hold:*

1. **Image of v_S :** $v_S(F_T) \subseteq F_{S \cap T}$.
2. **Eigenspace of v_S :** *If $T \subseteq S$, then $v_S f = f$ for all $f \in F_T$.*

This lemma is a straightforward consequence of the definitions of value operators and determined function spaces. We defer a proof to Appendix C.1.

Lemma 13 shows that the operator v_S essentially projects all determined function spaces into their image within F_S . Furthermore, it implies Properties 1 and 2 of Lemma 10. Property 1 holds

since *any* value operator must map $F_{[d] \setminus \{i\}}$ to itself meaning the same holds for any linear combination of value operators (such as A_i). Property 2 holds since the specific value operators that comprise B_i all corresponds to subsets of $[d] \setminus \{i\}$, which means their images must all be constrained to $F_{[d] \setminus \{i\}}$.

5.2. The Shapley Lie Algebra

We begin by defining the Shapley Lie algebra.

Definition 14 (Shapley Lie Algebra) *The Shapley Lie algebra \mathfrak{g}_Φ is the Lie algebra generated by $\{v_S : S \subseteq [d]\}$. That is, \mathfrak{g}_Φ is the smallest set of linear operators containing all v_S such that for all $v, w \in \mathfrak{g}_\Phi$,*

1. $\forall a, b \in \mathbb{C}, av + bw \in \mathfrak{g}_\Phi$,
2. $[v, w] = vw - wv$ is also in \mathfrak{g}_Φ .

The operation $[v, w]$ is called the *derivation* of v and w . Observe that v, w commute if and only if $[v, w] = 0$. More broadly, $[v, w]$ can be thought of as representing the degree to which v and w commute. This idea is expressed through *solvability*, which can be thought of as a generalization of commutativity. We now state the main result of this section, that the Shapley Lie algebra is solvable.

Lemma 15 (Shapley Lie Algebra is Solvable) *The Shapley Lie algebra is solvable, meaning that the following holds:*

1. For any Lie algebra, \mathfrak{g} , its derivation $[\mathfrak{g}, \mathfrak{g}]$ is the Lie sub-algebra generated by $\{[v_1, v_2] : v_1, v_2 \in \mathfrak{g}\}$.
2. Define \mathfrak{g}_Φ 's derived series is the sequence $\mathfrak{g}_\Phi^{(i)} = [\mathfrak{g}_\Phi^{(i-1)}, \mathfrak{g}_\Phi^{(i-1)}]$ with $\mathfrak{g}_\Phi^{(0)} = \mathfrak{g}_\Phi$.
3. Then there exists n such that $\mathfrak{g}_\Phi^{(n)} = 0$.

The length of a Lie algebra's derived series serves as a measure of how “far” the algebra is from being commutative.

To prove Lemma 15, we begin by restricting our attention to elements of \mathfrak{g}_Φ that can be constructed from value operators *purely* through derivations, rather than through derivations and linear combinations. Such elements will prove easier to analyze using Lemma 13.

Definition 16 (Pure operators) *The set of pure operators in \mathfrak{g}_Φ is defined as the smallest subset $V \subseteq \mathfrak{g}_\Phi$ that is closed under derivations and also contains all value operators.*

The key idea to showing that \mathfrak{g}_Φ is solvable is to generalize Lemma 13 to apply to pure operators.

Lemma 17 (Images and Eigenspaces of Pure Operators) *Let $V \cap \mathfrak{g}_\Phi^{(1)}$ denote all pure operators that are in the derived Lie sub-algebra of \mathfrak{g}_Φ . For all pure operators $v \in V \cap \mathfrak{g}_\Phi^{(1)}$, there exists a subset $\alpha(v) \subseteq [d]$ such that the following hold.*

1. **Image of v :** $v(F_T) \subseteq F_{\alpha(v) \cap T}$.
2. **Eigenspace of v :** If $T \subseteq \alpha(v)$, then $vf = 0$ for all $f \in F_T$.
3. **Behavior with Derivations:** $\alpha([v, w]) \subseteq \alpha(v) \cap \alpha(w)$.

Observe that unlike Lemma 13, the eigenspace here operates with an eigenvalue of 0 rather than 1. This results from the way derivations are a difference between two operators. As a sanity check, when \mathfrak{g}_Φ is commutative, the lemma trivially holds with $\alpha(v) = \emptyset$ for all derived elements v .

The proof idea for Lemma 17 is to partition the set of pure operators into levels based upon how many derivations are needed to construct a given element (the value operators are on the 0th level). Then, beginning with Lemma 13, apply induction on the level. We defer a proof to Appendix C.3.

We now provide a proof sketch for Lemma 15.

Proof [Lemma 15 (Sketch); full proof in Appendix C.2] The key idea is to characterize the way applying derivations to pure elements effects the associated subset $\alpha(v)$. By repeatedly applying Lemma 17, we can show

$$\alpha([v, w]) \subseteq \alpha(v) \cap \alpha(w), \text{ and } \alpha(v) = \alpha(w) \implies [v, w] = 0.$$

Observe that together, these two statements imply that $\alpha([v, w])$ is strictly smaller in cardinality than $\max(|\alpha(v)|, |\alpha(w)|)$. From here it is relatively straightforward to see that applying $(d + 1)$ successive derivations to any set of pure operators will result in 0. To finish the result, we simply show (through checking definitions) that pure operators form bases (in the linear algebra sense) of all Lie sub-algebras in the sequence $\mathfrak{g}_\Phi^{(0)}, \mathfrak{g}_\Phi^{(1)}, \dots$. Thus, since the pure operators contained in $\mathfrak{g}_\Phi^{(d+1)}$ are 0, it follows that $\mathfrak{g}_\Phi^{(d+1)} = 0$ which implies solvability. ■

We conclude this section by sketching a proof of Property 3 of Lemma 10.

Proof [Property 3 of Lemma 10 (Sketch); full proof in Appendix A.3] As before, assume that F , the space over which all our operators act, is finite dimensional. Extending our argument to the infinite dimensional case is handled in the appendix. It then follows by an application of Lie’s theorem (Theorem 37) in conjunction with Lemma 15 that there exists a basis of F over which all operators in \mathfrak{g}_Φ are *simultaneously upper triangular*. Thus, $A_i = \frac{1}{d} \sum_{S \subseteq [d] \setminus \{i\}} \binom{d-1}{|S|}^{-1} v_{S \cup \{i\}}$ can be written as a strictly positive sum of upper triangular matrices, and is itself upper triangular. Finally, since $v_{[d]}$ is included in this sum, and since $v_{[d]}$ is the identity operator, it follows that A_i has a strictly positive diagonal which implies that it is invertible. ■

6. Discussion

The main goal of this work is to investigate the soundness of the widely used practice of aggregating SHAP values. We show that provided they are computed over the extended distribution, SHAP and KernelSHAP values can be used to soundly eliminate unimportant features. We stress that our results are not intended to suggest these algorithms as a first choice for eliminating features – there exist other better methods for doing so. We instead contend that our results guarantee soundness in settings where SHAP and KernelSHAP are being routinely applied — provided practitioners adopt our modification to aggregate over the extended data support. In practice, this modification is straightforward to implement and does not require to change the SHAP packages’ internal code. One only has to replace a sample from μ with a sample from μ^* , which can easily be achieved by scrambling data columns appropriately (see Algorithm 1).

Our techniques may also be useful for analyzing other properties of SHAP as well. As a testament to their versatility, they readily apply to both interventional SHAP values and KernelSHAP values, despite the latter lacking a formal connection to the former.

Acknowledgments

We thank Gunnar König for his helpful discussions on KernelSHAP and Sebastian Bordt for his feedback. This work has been supported by the German Research Foundation through the Cluster of Excellence “Machine Learning - New Perspectives for Science” (EXC 2064/1 number 390727645) and the Carl Zeiss Foundation through the CZS Center for AI and Law.

References

- M. Berdugo, J. J. Gaitán, M. Delgado-Baquerizo, T. W. Crowther, and V. Dakos. Prevalence and drivers of abrupt vegetation shifts in global drylands. *Proceedings of the National Academy of Sciences (PNAS)*, 119(43):e2123393119, 2022.
- D. Bernard, E. Doumard, I. Ader, P. Kemoun, J. Pagès, A. Galinier, S. Cussat-Blanc, F. Furger, L. Ferrucci, J. Aligon, et al. Explainable machine learning framework to predict personalized physiological aging. *Aging cell*, 22(8):e13872, 2023.
- B. Bilodeau, N. Jaques, P. W. Koh, and B. Kim. Impossibility theorems for feature attribution. *Proceedings of the National Academy of Sciences (PNAS)*, 121(2):e2304406120, 2024.
- S. Bordt and U. von Luxburg. From Shapley values to generalized additive models and back. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.
- M. Bressan, N. Cesa-Bianchi, E. Esposito, Y. Mansour, S. Moran, and M. Thiessen. A theory of interpretable approximations. In *Conference on Learning Theory (COLT)*, 2024.
- Z. Chen, C. Hou, L. Wang, C. Yu, T. Chen, B. Shen, Y. Hou, P. Li, and T. Li. Screening membraneless organelle participants with machine-learning models that integrate multimodal features. *Proceedings of the National Academy of Sciences (PNAS)*, 119(24):e2115369119, 2022.
- I. Covert and S.-I. Lee. Improving KernelSHAP: Practical Shapley value estimation via linear regression. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.
- I. Covert, S. M. Lundberg, and S.-I. Lee. Understanding global feature contributions with additive importance measures. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- S. Dasgupta, N. Frost, and M. Moshkovitz. Framework for evaluating faithfulness of local explanations. In *International Conference on Machine Learning (ICML)*, 2022.
- C. S. Delavaux, T. W. Crowther, C. M. Zohner, N. M. Robmann, T. Lauber, J. van den Hoogen, S. Kuebbing, J. Liang, S. de Miguel, G.-J. Nabuurs, et al. Native diversity buffers against severity of non-native tree invasions. *Nature*, 621(7980):773–781, 2023.
- I. U. Ekanayake, D. P. P. Meddage, and U. Rathnayake. A novel approach to explain the black-box nature of machine learning in compressive strength predictions of concrete using Shapley additive explanations (SHAP). *Case Studies in Construction Materials*, 16:e01059, 2022.
- F. K. Ewald, L. Bothmann, M. N. Wright, B. Bischl, G. Casalicchio, and G. König. A guide to feature importance methods for scientific inference. In *World Conference on Explainable Artificial Intelligence*, 2024.

- C. Frye, C. Rowat, and I. Feige. Asymmetric Shapley values: incorporating causal knowledge into model-agnostic explainability. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- O. Giuntella, K. Hyde, S. Saccardo, and S. Sadoff. Lifestyle and mental health disruptions during COVID-19. *Proceedings of the National Academy of Sciences (PNAS)*, 118(9):e2016632118, 2021.
- E. E. Greenwood, T. Lauber, J. van den Hoogen, A. Donmez, R. E. S. Bain, R. Johnston, T. W. Crowther, and T. R. Julian. Mapping safe drinking water use in low- and middle-income countries. *Science*, 385(6710):784–790, 2024.
- S. Hooker, D. Erhan, P.-J. Kindermans, and B. Kim. A benchmark for interpretability methods in deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
- S. Jiang, L. Tarasova, G. Yu, and J. Zscheischler. Compounding effects in flood drivers challenge estimates of extreme river floods. *Science Advances*, 10(13):eadl4005, 2024.
- I. E. Kumar, S. Venkatasubramanian, C. Scheidegger, and S. Friedler. Problems with Shapley-value-based explanations as feature importance measures. In *International Conference on Machine Learning (ICML)*, 2020.
- J. Lei, M. G’Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.
- J. Li, H. Nishikawa, J. Kougo, J. Zhou, S. Dai, W. Tang, X. Zhao, Y. Hisai, M. Huang, and S. Aya. Development of ferroelectric nematic fluids with giant- ϵ dielectricity and nonlinear optical properties. *Science Advances*, 7(17):eabf5047, 2021.
- S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee. From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1):56–67, 2020.
- W. E. Marcílio and D. M. Eler. From explanations to feature selection: assessing SHAP values as feature selection mechanism. In *Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2020.
- C. Martínez-Ruiz, J. R. M. Black, C. Puttick, M. S. Hill, J. Demeulemeester, E. Larose Cadieux, K. Thol, T. P. Jones, S. Veeriah, C. Naceur-Lombardelli, et al. Genomic–transcriptomic evolution in lung cancer and metastasis. *Nature*, 616(7957):543–552, 2023.
- L. Merrick and A. Taly. The explanation game: Explaining machine learning models using Shapley values. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction (CD-MAKE)*, 2020.
- C. Molnar. *Interpretable Machine Learning*. 2nd edition, 2022.

- A. Mor, Y. Belinkov, and B. Kimelfeld. Accelerating the global aggregation of local explanations. In *AAAI Conference on Artificial Intelligence*, 2024.
- S. Qiu, M. I. Miller, P. S. Joshi, J. C. Lee, C. Xue, Y. Ni, Y. Wang, I. De Anda-Duran, P. H. Hwang, J. A. Cramer, et al. Multimodal deep learning for Alzheimer’s disease dementia assessment. *Nature Communications*, 13(1):3404, 2022.
- R. P. Rane, E. F. de Man, J. Kim, K. Görgen, M. Tschorn, M. A. Rapp, T. Banaschewski, A. L. W. Bokde, S. Desrivieres, H. Flor, et al. Structural differences in adolescent brains can predict alcohol misuse. *eLife*, 11:e77545, 2022.
- M. T. Ribeiro, S. Singh, and C. Guestrin. ”Why should i trust you?”: Explaining the predictions of any classifier. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.
- M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI Conference on Artificial Intelligence*, 2018.
- L. S. Shapley. A value for n-person games. *Contribution to the Theory of Games*, 1953.
- M. Sharma Timilsina, S. Sen, B. Upreti, V. B. Patel, P. Sharma, and P. N. Sheth. Prediction of HHV of fuel by machine learning algorithm: Interpretability analysis using Shapley additive explanations (SHAP). *Fuel*, 357:129573, 2024.
- D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju. Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods. In *AAAI/ACM Conference on AI, Ethics, and Society*, 2020.
- M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning (ICML)*, 2017.
- I. van der Linden, H. Haned, and E. Kanoulas. Global aggregations of local explanations for black box models. *arXiv preprint arXiv:1907.03039*, 2019.
- I. Verdinelli and L. Wasserman. Feature importance: A closer look at Shapley values and LOCO. *Statistical Science*, 39(4):623–636, 2024.
- D. Wang, S. Thunéll, U. Lindberg, L. Jiang, J. Trygg, and M. Tysklind. Towards better process management in wastewater treatment plants: Process analytics based on SHAP values for tree-based machine learning methods. *Journal of Environmental Management*, 301:113941, 2022.
- J. T. Wang, T. Yang, J. Zou, Y. Kwon, and R. Jia. Rethinking data Shapley for data selection tasks: Misleads and merits. In *International Conference on Machine Learning (ICML)*, 2024.
- A. Wojtuch, R. Jankowski, and S. Podlowska. How can SHAP values help to shape metabolic stability of chemical compounds? *Journal of Cheminformatics*, 13(74), 2021.
- J. Yang, L. Tao, J. He, J. R. McCutcheon, and Y. Li. Machine learning enables interpretable discovery of innovative polymers for gas separation membranes. *Science Advances*, 8(29):eabn9545, 2022.

Appendix A. Proofs from Section 3

A.1. Proof of Theorem 6

Recall that in our proof sketch, we treated F as though it were finite dimensional. To handle the infinite case, we will apply Lemma 41, that allows us to restrict our attention to a finite dimensional subspace $F_f \subseteq F$. A statement and proof of Lemma 41 can be found in Section E of the appendix. We are now prepared to prove Theorem 6.

Proof [Theorem 6] It suffices to show that for all $f \in F$ and $1 \leq i \leq d$, $\Phi_i f = 0$ if and only if $f \in F_{[d] \setminus \{i\}}$.

(\Rightarrow) Suppose $\Phi_i f = 0$ which means that $A_i f = B_i f$. Property 2 of Lemma 10 implies $B_i f \in F_{[d] \setminus \{i\}}$, and thus $A_i f \in F_{[d] \setminus \{i\}}$. Our main idea will be to use a dimension counting argument to show that because A_i is invertible (Property 3 of Lemma 10) and because it preserves $F_{[d] \setminus \{i\}}$ (Property 1 of Lemma 10), f must itself be inside $F_{[d] \setminus \{i\}}$. This would be immediate from the Rank-Nullity theorem if $F_{[d] \setminus \{i\}}$ was finite dimensional. Unfortunately this is not quite the case as spaces of functions are typically infinite dimensional.

To circumvent this issue, we use Lemma 41 which states that F_f is both A_i and B_i -invariant. Let F_f be as defined in Definition 39 and let $W = F_f \cap F_{[d] \setminus \{i\}}$. Let W' be the vector space generated by W and f . Then by restricting A_i to W' and by using the fact that F_f is A_i -invariant along with property 1 of Lemma 10, we have $A_i(W') \subseteq W$. This now precisely corresponds to the finite dimensional case, and the Rank-Nullity theorem again implies $W' = W$ which means $f \in W \subseteq F_{[d] \setminus \{i\}}$ as desired.

(\Leftarrow) Suppose $f \in F_{[d] \setminus \{i\}}$. By directly utilizing the definition of SHAP values (Definition 2), for all $x \in \text{supp}(\mu^*)$ we have

$$\begin{aligned} \phi_i(\mu, f, x) &= \frac{1}{d} \sum_{S \subseteq [d] \setminus \{i\}} \binom{d-1}{|S|}^{-1} (v_{S \cup \{i\}}(f, x) - v_S(f, x)). \\ &= \frac{1}{d} \sum_{S \subseteq [d] \setminus \{i\}} \binom{d-1}{|S|}^{-1} (\mathbb{E}_{X \sim \mu}[f(x_{S \cup \{i\}}, X_{S^c \setminus \{i\}})] - \mathbb{E}_{X \sim \mu}[f(x_S, X_{S^c})]) \\ &= \frac{1}{d} \sum_{S \subseteq [d] \setminus \{i\}} \binom{d-1}{|S|}^{-1} \mathbb{E}_{X \sim \mu}[f(x_S, x_{\{i\}}, X_{S^c \setminus \{i\}}) - f(x_S, X_{\{i\}}, X_{S^c \setminus \{i\}})]. \end{aligned}$$

Observe that the difference above is taken over f evaluated at two points that only differ in their i th coordinate. Since f is $[d] \setminus \{i\}$ -determined, this must equal 0 which implies the result. \blacksquare

A.2. Proof of Lemma 8

Proof [Lemma 8] We first show it is well defined. The only cause for concern is that $v_S(\mu, f, x)$ was defined over functions $f : \mathbb{R}^d \rightarrow \mathbb{R}$ rather than $f : \text{supp}(\mu^*) \rightarrow \mathbb{R}$. To resolve this, we expand out $v_S(\mu, f, x)$ using the definition of a value function (Definition 1). We have,

$$(v_S f)(x) = v_S(\mu, f, x) = \mathbb{E}_{X \sim \mu}[f(x_S, X_{S^c})].$$

For any i , x_i and X_i lie within the support of μ_i^* (Definition 5) and thus $(x_S, X_{S^c})_i$ does as well. Since this holds for all i , it follows $(x_S, X_{S^c})_i \in \text{supp}(\mu_i^*)$ which implies $f(x_S, X_{S^c})$ is well defined.

Finally, the fact that v_S is linear is an immediate consequence of the linearity of the expectation which concludes the proof. \blacksquare

A.3. Proof of Lemma 10

Recall that our strategy is to utilize Lemma 13 to prove Properties 1 and 2, and Lemma 15 along with Lie's theorem (Theorem 37) to prove Property 3. To do so, we begin by first proving a key technical lemma that will enable us to not only derive Property 3, but also assist in proving Theorem 11 (given in Section 3.4).

Lemma 18 (Eigenvalues of A_i) *Let $f \in F$ be a non-zero function such that $A_i f = \lambda f$ for some $\lambda \in \mathbb{C}$. Then λ is a positive real number with $\lambda \geq \frac{1}{d}$.*

Proof [Lemma 18] Fix any $f \in F$ with $A_i f = \lambda f$ for $\lambda \in \mathbb{C}$. By Lemma 40, there exists a finite dimensional representation (see Definition 36) (ρ_f, F_f) of \mathfrak{g}_Φ such that

1. $f \in F_f$,
2. for all $v \in \mathfrak{g}_\Phi$, $\rho_f(v) : F_f \rightarrow F_f$ is the restriction of v to F_f . In particular, this means F_f is a v -invariant subspace for all $v \in \mathfrak{g}_\Phi$.

By Lemma 15, \mathfrak{g}_Φ is a solvable Lie algebra. Thus, we can apply Lie's theorem (Theorem 37), which implies that there exists a basis of F_f over which all represented value operators $\rho_f(v_S)$ can be expressed as upper triangular matrices M_S . Furthermore, Lemma 13 implies that for all $S \subseteq [d]$, $v_S v_S = v_S$. Thus, all of the eigenvalues of v_S are either 0 or 1, which implies diagonal elements of M_S are either 0 or 1 as well.

Since $A_i \in \mathfrak{g}_\Phi$, we see that A_i itself can be represented as the matrix

$$M(A_i) = \frac{1}{d} \sum_{S \subseteq [d] \setminus \{i\}} \binom{d-1}{|S|}^{-1} M_{S \cup \{i\}}.$$

This matrix too is upper triangular, and its diagonal elements are all positive linear combinations of elements that are either 0 or 1. Thus, all diagonal elements of $M(A_i)$ are nonnegative real numbers. Furthermore, since $M_{[d]}$ itself is included in this sum, and since $v_{[d]}$ is simply the identity operator, it follows that *every diagonal element* of $M(A_i)$ is at least $\frac{1}{d} \binom{d-1}{d-1}^{-1} = \frac{1}{d}$.

Finally, since $M(A_i)$ is upper triangular, its diagonal elements are precisely its eigenvalues. However, f is included in F_f and satisfies $A_i f = \lambda f$. This implies $M(A_i) f = \lambda f$. Since f is non-zero, λ is an eigenvalue of $M(A_i)$. Thus λ is a diagonal element of $M(A_i)$ and is a real number that is at least $\frac{1}{d}$, as desired. \blacksquare

We are now prepared to prove Lemma 10.

Proof [Lemma 10] Observe that A_i and B_i are both linear combinations of value functions, and thus elements of \mathfrak{g}_Φ . Thus, applying Property 1 of Lemma 13 implies that $A_i(F_S) \subseteq F_S$ for all S (Property 1). Meanwhile, for all $S \subseteq [d] \setminus \{i\}$, the lemma similarly implies

$$v_S(F) \subseteq F_{S \cap [d]} = F_S \subseteq F_{[d] \setminus \{i\}},$$

with the last inclusion holding since S -determined functions are clearly $[d] \setminus \{i\}$ -determined as $S \subseteq [d] \setminus \{i\}$. Summing those over the definition of B_i implies $B_i(F) \subseteq F_{[d] \setminus \{i\}}$ (Property 2).

Finally, we prove Property 3. Fix $f \in F$ with $A_i f = 0$. Then Lemma 18 implies that f must be 0 as otherwise 0 would be an eigenvalue of A_i that is smaller than $\frac{1}{d}$. This implies Property 3. ■

A.4. Proof of Theorem 11

We begin by precisely defining the operators that correspond to computing SHAP values over the *entire* extended distribution (Definition 5).

Definition 19 (Value and SHAP Operators) *Let μ be a distribution over \mathbb{R}^d and μ^* its extended distribution. Let F be the space of all measurable functions $\text{supp}(\mu^*) \rightarrow \mathbb{C}$. For $S \subseteq [d]$, we define the value operator v_S^* as*

$$v_S^* f(x) = \mathbb{E}_{X \sim \mu^*} [f(x_S, X_{S^c})].$$

We then define the SHAP operator Φ_i^ for $1 \leq i \leq d$ with*

$$\Phi_i^* f = A_i^* f - B_i^* f = \frac{1}{d} \sum_{S \subseteq [d] \setminus \{i\}} \binom{d-1}{|S|}^{-1} v_{S \cup \{i\}}^* f - \binom{d-1}{|S|}^{-1} v_S^* f.$$

The definitions of v_S^* , A_i^* , B_i^* and Φ_i^* all directly correspond to Definitions 8 and 9. More generally, we will use the $*$ notation to denote the analog of quantity when μ is replaced by μ^* . The only thing to note is that the spaces of determined functions (Definition 7) we operate on, $F_S : S \subseteq [d]$, *remain unchanged*. This is because the extended support of μ^* is simply itself, as $(\mu^*)^* = \mu^*$. This is clear from the definition of μ^* .

To avoid confusion, **we will use the $*$ notation in all cases *except* determined function spaces, where we will omit the $*$.**

Next, we turn our attention towards the main business of this section which is proving Theorem 11. To this end, we define an inner product over F based on μ^* .

Definition 20 (Inner product with μ^*) *For $f, g \in F$, we define*

$$\langle f, g \rangle = \int_{\text{supp}(\mu^*)} \overline{f(x)} g(x) d\mu^*(x),$$

where μ^ denotes the extended distribution of μ .*

Here, \bar{z} denotes the complex conjugate of $z \in \mathbb{C}$. It can be easily verified that this is a well-defined inner product that makes $(F, \langle -, - \rangle)$ a Hilbert space. Next, for $S \subseteq [d]$, recall that v_S^* denotes the value operator taken with respect to μ^* . That is,

$$v_S^* f(x) = \mathbb{E}_{X \sim \mu^*} [f(x_S, X_{S^c})].$$

The key idea for eventually proving Theorem 11 is to show that v_S^* is Hermitian:

Lemma 21 (Value operators over μ^* are Hermitian) *For all $S \subseteq [d]$, v_S^* is Hermitian with respect to the inner product given in Definition 20. That is, for all $f, g \in F$,*

$$\langle v_S^* f, g \rangle = \langle f, v_S^* g \rangle.$$

Proof [Lemma 21] Our main idea is to exploit the fact that μ^* is a product of d independent distributions, μ_1^*, \dots, μ_d^* . For any $T \subseteq [d]$, let $\mu_T^* = \prod_{i \in T} \mu_i^*$. It follows that $\mu^* = \mu_T^* \times \mu_{T^c}^*$. To help simplify notation, for $p \in \text{supp}(\mu_T^*)$ and $q \in \text{supp}(\mu_{T^c}^*)$, we let $f(p, q)$ denote $f(x)$ where $x_T = p$ and $x_{T^c} = q$.

Applying this along with the definition of value operators, we see that

$$\begin{aligned}
 \langle v_S^* f, g \rangle &= \int_{\text{supp}(\mu^*)} \overline{(v_S^* f)(x)} g(x) d\mu^*(x) \\
 &= \int_{\text{supp}(\mu^*)} \left(\int_{\text{supp}(\mu^*)} \overline{f(x_S, X_{S^c})} d\mu^*(X) \right) g(x) d\mu^*(x) \\
 &= \int_{\text{supp}(\mu_S^*) \times \text{supp}(\mu_{S^c}^*)} \left(\int_{\text{supp}(\mu_{S^c}^*)} \overline{f(p, q)} d\mu_{S^c}^*(q) \right) g(p, r) d\mu_S^*(p) d\mu_{S^c}^*(r) \\
 &= \int_{\text{supp}(\mu_S^*) \times \text{supp}(\mu_{S^c}^*) \times \text{supp}(\mu_{S^c}^*)} \overline{f(p, q)} g(p, r) d\mu_S^*(p) d\mu_{S^c}^*(q) d\mu_{S^c}^*(r) \\
 &= \int_{\text{supp}(\mu_S^*) \times \text{supp}(\mu_{S^c}^*)} \left(\int_{\text{supp}(\mu_{S^c}^*)} g(p, r) d\mu_{S^c}^*(r) \right) \overline{f(p, q)} d\mu_S^*(p) d\mu_{S^c}^*(q) \\
 &= \int_{\text{supp}(\mu^*)} \left(\int_{\text{supp}(\mu^*)} g(x_S, X_{S^c}) d\mu^*(X) \right) \overline{f(x)} d\mu^*(x) \\
 &= \int_{\text{supp}(\mu^*)} (v_S^* g)(x) \overline{f(x)} d\mu^*(x) = \langle f, v_S^* g \rangle
 \end{aligned}$$

Basically, expanding out the inner product gives an integral over 2 sets of variables, one drawn from μ^* corresponding to the expectation, and another drawn from $\mu_{S^c}^*$ corresponding to the value operator. Due to the independent nature of these variables, they can be freely reordered resulting in the manipulation above. \blacksquare

Next, we show how to relate $\overline{\phi_i}(\mu^*, f)$, which is related to the absolute value of f , to the norm, $\langle \Phi_i^* f, \Phi_i^* f \rangle$.

Lemma 22 (Bounding Norm with Aggregate SHAP) *Let $f \in F$ be a function such that $f(x) \in [0, 1]$ for all $x \in \text{supp}(\mu^*)$. Then for all $1 \leq i \leq d$, $\langle \Phi_i^* f, \Phi_i^* f \rangle \leq \overline{\phi_i}(\mu^*, f)$.*

Proof [Lemma 22] Recall that Φ_i^* is an operator that maps F to itself. We begin by bounding the range of $\Phi_i^* f$. To do so, for any $S \subseteq [d]$, observe that for all $x \in \text{supp}(\mu^*)$,

$$(v_S^* f)(x) = \mathbb{E}_{X \sim \mu^*} [f(x_S, X_{S^c})] \in [0, 1],$$

Since everything within the expectation is an application of f which has range in $[0, 1]$. It immediately follows that $A_i^* f, B_i^* f \geq 0$, as A_i^*, B_i^* are both positive linear combinations of value operators

(Definition 19). To get upper bounds on the range of these functions, we see that

$$\begin{aligned}
(A_i^* f)(x) &= \frac{1}{d} \sum_{S \subseteq [d] \setminus \{i\}} \binom{d-1}{|S|}^{-1} (v_{S \cup \{i\}}^* f)(x) \\
&\leq \frac{1}{d} \sum_{S \subseteq [d] \setminus \{i\}} \binom{d-1}{|S|}^{-1} \\
&= \frac{1}{d} \sum_{j=0}^{d-1} \binom{d-1}{j}^{-1} \sum_{S \subseteq [d] \setminus \{i\}, |S|=j} 1 \\
&= \frac{1}{d} \sum_{j=0}^{d-1} \binom{d-1}{j}^{-1} \binom{d-1}{j} \\
&= 1.
\end{aligned}$$

An analogous argument shows $(B_i^* f)(x) \leq 1$. It follows that $(\Phi_i^* f)(x) = (A_i^* f)(x) - (B_i^* f)(x)$ must be an element in $[-1, 1]$. Substituting this, we find that

$$\begin{aligned}
\overline{\phi_i}(\mu^*, f) &= \int_{\text{supp}(\mu^*)} |(\Phi_i^* f)(x)| d\mu^*(x) \\
&\geq \int_{\text{supp}(\mu^*)} |(\Phi_i^* f)(x)|^2 d\mu^*(x) \\
&= \langle \Phi_i^* f, \Phi_i^* f \rangle.
\end{aligned}$$

■

We are now prepared to prove Theorem 11.

Proof [Theorem 11]

let F_f^* be the finite dimensional subspace defined in Lemma 41 that corresponds to μ^* (the subspace in the lemma was defined for an arbitrary measure μ). Let $W = F_f^* \cap F_{[d] \setminus \{i\}}$. Then Lemmas 41 and 10 imply that

1. $A_i^*(F_f^*) \subseteq F_f^*$,
2. $A_i^*(W) \subseteq W$,
3. $B_i^*(F_f^*) \subseteq W$,
4. $(A_i^*)^{-1}(\{0\}) = \{0\}$.

Let a_i^* and b_i^* denote the restrictions of A_i^* and B_i^* to F_f^* . It follows that these too are well defined operators that map $F_f^* \rightarrow F_f^*$, and also satisfy

$$(a_i^* - b_i^*)(h) = (A_i^* - B_i^*)(h) = \Phi_i^* h,$$

for all $h \in F_f^*$. Since F_f^* is finite dimensional, it follows that a_i^* has an inverse, $(a_i^*)^{-1}$.

Next, Lemma 21 implies that every value operators, v_S^* is Hermitian, which implies that A_i^* must be as well (as it is a linear combination of value operators). Since $W \subseteq F_f^* \subseteq F$, they inherit the inner product structure from F , and it follows that a_i^* and $(a_i^*)^{-1}$ are Hermitian as well. Since A_i^* has real eigenvalues that are all at least $\frac{1}{d}$ (Lemma 18), it follows that $(a_i^*)^{-1}$ has maximum eigenvalue at most d . It follows by standard linear algebra that for all $h \in F_f^*$,

$$\langle (a_i^*)^{-1}h, (a_i^*)^{-1}h \rangle \leq d^2 \langle h, h \rangle. \quad (2)$$

We are finally ready to prove Theorem 11. We claim $g = (a_i^*)^{-1}b_i^*f$ suffices. Observe that this is well defined as $f \in F_f^*$ (Lemma 41) and $a_i^*, (a_i^*)^{-1}$, and b_i^* are all well defined over this space.

To show that g suffices, we must show that $g \in F_{[d] \setminus \{i\}}$ and that $\langle f - g, f - g \rangle \leq d\epsilon$. For the first claim, we apply the 4 properties that we derived at the beginning of this proof. First, $b_i^*f = B_i^*f \in W$ by Property 3. Second, Properties 2 and 4 imply that W is an a_i^* invariant subspace. Since W is finite dimensional and since a_i^* is invertible, it follows that W is also $(a_i^*)^{-1}$ invariant. Thus $(a_i^*)^{-1}b_i^*f \in W$. This implies $g \in W \subseteq F_{[d] \setminus \{i\}}$, as desired.

For the second claim, we use the fact that $\overline{\phi}_i(\mu^*, f) \leq \epsilon$. By Lemma 22, this implies that $\langle \Phi_i^*f, \Phi_i^*f \rangle \leq \epsilon$. Using this, we see that

$$\begin{aligned} \int_{\mathbb{R}^d} (f(x) - g(x))^2 d\mu^*(x) &= \langle f - g, f - g \rangle \\ &= \langle f - (a_i^*)^{-1}b_i^*f, f - (a_i^*)^{-1}b_i^*f \rangle \\ &= \langle (a_i^*)^{-1}(a_i^*f - b_i^*f), (a_i^*)^{-1}(a_i^*f - b_i^*f) \rangle \\ &\leq d^2 \langle a_i^*f - b_i^*f, a_i^*f - b_i^*f \rangle \\ &= d^2 \langle \Phi_i^*f, \Phi_i^*f \rangle \leq d^2\epsilon. \end{aligned}$$

Here we simply substitute Equation 2 to simplify the inner product. This completes the proof. \blacksquare

Appendix B. Proofs from Section 4

B.1. Preliminaries on KernelSHAP

The main idea of KernelSHAP (Lundberg and Lee, 2017) is to use linear regression to *approximate* the value function. To construct KernelSHAP values at a point x , KernelSHAP learns a weight vector, $\mathcal{K}^x \in \mathbb{R}^d$, such that for all $S \subseteq [d]$,

$$v_S(\mu, f, x) \approx \langle \mathcal{K}^x, \mathbf{1}_S \rangle + \mathbb{E}_{X \sim \mu}[f(X)], \quad (3)$$

where $\mathbf{1}_S$ is an indicator vector for S with $(\mathbf{1}_S)_i = \mathbb{1}(i \in S)$. The key observation is that if this approximation was a precise equality, then

$$\begin{aligned}
\phi_i(\mu, f, x) &= \frac{1}{d} \sum_{S \subseteq [d] \setminus \{i\}} \binom{d-1}{|S|}^{-1} (v_{S \cup \{i\}}(\mu, f, x) - v_S(\mu, f, x)) \\
&= \frac{1}{d} \sum_{S \subseteq [d] \setminus \{i\}} \binom{d-1}{|S|}^{-1} (\langle \mathcal{K}^x, \mathbf{1}_{S \cup \{i\}} \rangle + \mathbb{E}_{X \sim \mu}[f(x)] - \langle \mathcal{K}^x, \mathbf{1}_S \rangle - \mathbb{E}_{X \sim \mu}[f(x)]) \\
&= \frac{1}{d} \sum_{S \subseteq [d] \setminus \{i\}} \binom{d-1}{|S|}^{-1} \langle \mathcal{K}^x, \mathbf{1}_{S \cup \{i\}} - \mathbf{1}_S \rangle \\
&= \frac{1}{d} \sum_{S \subseteq [d] \setminus \{i\}} \binom{d-1}{|S|}^{-1} \mathcal{K}_i^x = \mathcal{K}_i^x.
\end{aligned}$$

For this reason, KernelSHAP simply outputs the coefficients of the linear regression as its approximations to the SHAP values.

KernelSHAP solves the linear regression suggested in Equation 3 by using OLS as follows. Let $X = \{x^{(1)}, \dots, x^{(n)}\} \sim \mu^n$ be an i.i.d sample from μ , $Z = \{S_1, \dots, S_n\} \sim \pi^n$ be an i.i.d sample of points from a probability distribution π over $[d]$ defined by

$$\pi(S) \propto \begin{cases} \frac{d-1}{\binom{d}{|S|}|S|(d-|S|)} & 0 < |S| < d \\ 0 & \text{otherwise} \end{cases}. \quad (4)$$

Here, Z is a set of i.i.d subsets from $[d]$ that are weighted in likelihood according to how prevalent they are in the weighting scheme used in the definition of SHAP. Note that the weights for sets of size d and 0 is 0 – this is because the values of $\langle \mathcal{K}^x, \mathbf{1}_{[d]} \rangle$ and $\langle \mathcal{K}^x, \mathbf{1}_\emptyset \rangle$ are enforced directly using constraints as their corresponding value functions can be more precisely estimated. It is for this reason that the weights given by π_S do not *exactly* match the weights that appear in Definition 2. For a more extended discussion of this choice, see [Lundberg and Lee \(2017\)](#).

Putting it all together, KernelSHAP solves the following OLS regression. For $x \in \mathbb{R}^d$, $f : \mathbb{R}^d \rightarrow \mathbb{R}$, and X, Z sampled as above,

$$\begin{aligned}
(\mathcal{K}_1(X, f, x), \dots, \mathcal{K}_d(X, f, x)) &= \arg \min_{\mathcal{K}^x \in \mathbb{R}^d} \frac{1}{n} \sum_{j=1}^n \left(f(x_{S_j}, x_{S_j^c}^{(j)}) - \langle \mathcal{K}^x, \mathbf{1}_{S_j} \rangle - \bar{f} \right)^2 \\
\text{such that } \bar{f} &= \frac{1}{n} \sum_{j=1}^n f(x^{(j)}) \\
\langle \mathbf{1}_{[d]}, \mathcal{K}^x \rangle &= f(x) - \bar{f}.
\end{aligned} \quad (5)$$

In their analysis of KernelSHAP, [Covert and Lee \(2021\)](#) provide explicits for $\mathcal{K}_i(X, f, x)$ and $\mathcal{K}_i(\mu, f, x)$, where the latter is a limit object that the former converges towards in the large sample limit (see Definition 24). Both of these expressions will be extremely useful in our analysis, and we include them here.

Lemma 23 (Solution to KernelSHAP: Equation 7 of Covert and Lee (2021)) Let $x \in \mathbb{R}^d$ be a point, $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be a function, $X = \{x^{(1)}, \dots, x^{(n)}\}$ a dataset, and $Z = \{S_1, \dots, S_n\} \sim \pi^n$ be a set of subsets drawn according to π (Equation 4). Then the solution to Equation 5 can be written as follows. Let $M_n \in \mathbb{R}^{d \times d}$ and $b_n \in \mathbb{R}^d$ be defined as

$$M_n = \frac{1}{n} \sum_{j=1}^n \mathbf{1}_{S_j} \mathbf{1}_{S_j}^t \text{ and } b_n = \frac{1}{n} \sum_{j=1}^n \mathbf{1}_{S_j} \left(f(x_{S_j}, x_{S_j^c}^{(j)}) - \bar{f} \right).$$

Then the vector $\mathcal{K}(X, f, x) = (\mathcal{K}_1(X, f, x), \dots, \mathcal{K}_d(X, f, x))$ is equal to

$$\mathcal{K}(X, f, x) = M_n^{-1} \left(b_n - \mathbf{1} \frac{\mathbf{1}^t M_n^{-1} b_n - f(x) + \bar{f}}{\mathbf{1}^t M_n^{-1} \mathbf{1}} \right),$$

where $\mathbf{1}$ is shorthand for the all ones vector, $\mathbf{1}_{[d]} \in \mathbb{R}^d$.

Definition 24 (Limit Object of KernelSHAP: Equation 8 of Covert and Lee (2021)) Let $x \in \mathbb{R}^d$ be a point. Let μ be a distribution over \mathbb{R}^d and $f : \mathbb{R}^d \rightarrow \mathbb{R}$ a function. Let $M \in \mathbb{R}^{d \times d}$ and $b \in \mathbb{R}^d$ be defined as

$$M = \mathbb{E}_{S \sim \pi} [\mathbf{1}_S \mathbf{1}_S^t] \text{ and } b = \mathbb{E}_{S \sim \pi} [\mathbf{1}_S (v_S(\mu, f, x) - v_{\emptyset}(\mu, f, x))].$$

For any $x \in \mathbb{R}^d$, the limit object of KernelSHAP, $\mathcal{K}(\mu, f, x) = (\mathcal{K}_1(\mu, f, x), \dots, \mathcal{K}_d(\mu, f, x))$, is defined as:

$$\mathcal{K}(\mu, f, x) = M^{-1} \left(b - \mathbf{1} \frac{\mathbf{1}^t M^{-1} b - f(x) + v_{\emptyset}(\mu, f, x)}{\mathbf{1}^t M^{-1} \mathbf{1}} \right).$$

We now define the error term that represents how quickly KernelSHAP converges. For our purposes, we are interested in the behavior of KernelSHAP when it is run on X^* , which is the data matrix obtained by scrambling the columns of X (see Section 4).

Definition 25 (Error Term) We define the error term between the empirical computation of KernelSHAP and its limit object as follows:

$$\eta(X^*, \mu^*, f) = \max_{1 \leq i \leq d} \left| \left(\frac{1}{n} \sum_{x^{(j)} \in X^*} |\mathcal{K}_i(X^*, f, x^{(j)})| \right) - \mathbb{E}_{x \sim \mu^*} |\mathcal{K}_i(\mu, f, x)| \right|.$$

While it is clear from the law of large numbers that $\eta(X^*, \mu^*, f) \rightarrow 0$ as $n \rightarrow \infty$, the precise rate of convergence isn't clear. This problem is extensively studied in Covert and Lee (2021), where they give some rates of convergence along with plenty of empirical evidence that this rate is very fast in practice. For this reason, we express our result in terms of $\eta(X^*, \mu^*, f)$.

Finally, we conclude this section by citing the following useful explicit formula for M (given by Covert and Lee (2021)).

Lemma 26 (Explicit formula for M : Covert and Lee (2021)) Let I_d denote the $d \times d$ identity matrix, and J_d denote the $d \times d$ matrix consisting of all ones. Then $M = pI_d + qJ_d$ where

$$p = \frac{1}{2} - q \text{ and } q = \frac{1}{d(d-1)} \frac{\sum_{k=2}^{d-1} \frac{d-1}{d-k}}{\sum_{k=1}^{d-1} \frac{1}{k(d-k)}}.$$

B.2. Proof of Theorem 12

Let μ be fixed. We begin by defining an operator that corresponds to the limit object of KernelSHAP when taken over μ^* . As before, we let F denote the space of all functions $\text{supp}(\mu^*) \rightarrow \mathbb{C}$.

To simplify our algebra, we will also simply assume that $v_\emptyset^* f = 0$ – this can be accomplished by simply subtracting the mean of f from it. Doing so does not effect any of the SHAP values (and is in fact a common preprocessing step).

Definition 27 For $1 \leq i \leq d$, let $\mathcal{K}_i^* : F \rightarrow F$ be defined as the operator with $\mathcal{K}_i^* f(x) = \mathcal{K}_i(\mu^*, f, x)$.

Our main idea will be to rewrite \mathcal{K}_i^* as a linear combination of value operators. This will allow us to apply the same techniques that we've used to prove Theorems 6 and Theorem 11.

To do so, we will need several bits of algebra, which we break into the following lemmas. We will also let b_x denote the value of b that corresponds to x , as b was defined with respect to a point x (Definition 24).

Lemma 28 $\mathbf{1}^t M^{-1} b_x = \sum_{0 < |S| < d} \frac{|S|\pi(S)}{p+dq} v_S^* f(x)$.

Proof We just brute force it out. We have

$$\begin{aligned} \mathbf{1}^t M^{-1} b_x &= ((M^{-1})^t \mathbf{1})^t b_x \\ &= \frac{1}{p+dq} \mathbf{1}^t b_x \\ &= \frac{1}{p+dq} \mathbf{1}^t \sum_{0 < |S| < d} \mathbf{1}_S \pi(S) v_S(\mu^*, f, x) \\ &= \sum_{0 < |S| < d} \frac{|S|\pi(S)}{p+dq} v_S^* f(x), \end{aligned}$$

where in the middle step we simply used the fact that $\mathbf{1}$ is clearly an eigenvector of M with eigenvalue $p+dq$, along with the the observation that $\mathbf{1}^t \mathbf{1}_S = |S|$. \blacksquare

Lemma 29 For $1 \leq i \leq d$, $(M^{-1} b_x)_i = \sum_{S \subseteq [d] \setminus \{i\}} \alpha_{S \cup \{i\}} v_{S \cup \{i\}}^* f(x) - \beta_S v_S^* f(x)$, where

$$\alpha_{S \cup \{i\}} = \left(\frac{1}{p} - \frac{q|S \cup \{i\}|}{p(p+dq)} \right) \pi(S \cup \{i\}) \text{ and } \beta_S = \left(\frac{q|S|}{p(p+dq)} \right).$$

Proof With some more brute force,

$$\begin{aligned} M^{-1} b_x &= (pI_d + qJ_d)^{-1} \sum_{S \subseteq [d]} \mathbf{1}_S \pi(S) v_S(\mu^*, f, x) \\ &= \left(\frac{1}{p} I_d - \frac{q}{p(p+dq)} J_d \right) \sum_{S \subseteq [d]} \mathbf{1}_S \pi(S) v_S(\mu^*, f, x) \\ &= \left(\frac{1}{p} I_d - \frac{q}{p(p+dq)} J_d \right) \sum_{S \subseteq [d] \setminus \{i\}} \left(\mathbf{1}_{S \cup \{i\}} \pi(S \cup \{i\}) v_{S \cup \{i\}}^* f(x) + \mathbf{1}_S \pi(S) v_S^* f(x) \right). \end{aligned}$$

Now, observing that $(\mathbf{1}_{S \cup \{i\}})_i = 1$ and $(\mathbf{1}_S)_i = 0$ if $i \notin S$, we see that

$$\begin{aligned} \left(\frac{1}{p} I_d - \frac{q}{p(p+qd)} J_d \right) \sum_{S \subseteq [d] \setminus \{i\}} \mathbf{1}_{S \cup \{i\}} \pi(S \cup \{i\}) v_{S \cup \{i\}}^* f(x) \\ = \sum_{S \subseteq [d] \setminus \{i\}} \left(\frac{1}{p} - \frac{q|S \cup \{i\}|}{p(p+dq)} \right) \pi(S \cup \{i\}) v_{S \cup \{i\}}^* f(x) \end{aligned}$$

and

$$\begin{aligned} \left(\frac{1}{p} I_d - \frac{q}{p(p+qd)} J_d \right) \sum_{S \subseteq [d] \setminus \{i\}} \mathbf{1}_S \pi(S) v_S^* f(x) \\ = \sum_{S \subseteq [d] \setminus \{i\}} \left(-\frac{q|S|}{p(p+dq)} \right) \pi(S) v_S^* f(x) \end{aligned}$$

■

Lemma 30 (Expressing \mathcal{K}_i using value operators) *Let $1 \leq i \leq d$. There exist real numbers ι_S for all $S \subseteq [d]$ such that the following hold:*

1. $\mathcal{K}_i^* = \sum_{S \subseteq [d]} \iota_S v_S^*$.
2. $\iota_S \geq 0$ for all S where $i \in S$.
3. $\iota_{[d]} = \frac{1}{d}$.

Proof We explicitly compute $\mathcal{K}_i^* f(x) = \mathcal{K}_i(\mu^*, f, x)$. When possible we simplify using the two lemmas above. Using that $v_\emptyset^* f = 0$ and that $v_{[d]}^*$ is the identity map, we have

$$\begin{aligned} \mathcal{K}_i^* f(x) &= \left(M^{-1} \left(b_x - \mathbf{1} \frac{\mathbf{1}^t M^{-1} b_x - f(x) + v_\emptyset(\mu^*, f, x)}{\mathbf{1}^t M^{-1} \mathbf{1}} \right) \right)_i \\ &= \left(M^{-1} b_x - \frac{M^{-1} \mathbf{1}}{\mathbf{1}^t M^{-1} \mathbf{1}} (\mathbf{1}^t M^{-1} b_x - f(x)) \right)_i \\ &= \left(M^{-1} b_x - \frac{\mathbf{1}}{d} (\mathbf{1}^t M^{-1} b_x - v_{[d]}^* f(x)) \right)_i \\ &= \sum_{S \subseteq [d] \setminus \{i\}} \alpha_{S \cup \{i\}} v_{S \cup \{i\}}^* f(x) - \beta_S v_S^* f(x) + \frac{1}{d} v_{[d]}^* f(x) - \sum_{S \subseteq [d]} \frac{|S| \pi(S)}{d(p+dq)} v_S^* f(x), \end{aligned}$$

with the last step coming from substituting Lemmas 28 and 29. Regrouping terms, we see that

$$\iota_S = \begin{cases} \alpha_S - \frac{|S| \pi(S)}{d(p+dq)} & i \in S, |S| < d \\ -\beta_S - \frac{|S| \pi(S)}{d(p+dq)} & i \notin S, |S| < d \\ \frac{1}{d} & S = [d] \end{cases}$$

Properties 1 and 3 of the lemma clearly hold, so all that is left is Property 2. To this end, we see that for $i \in S$,

$$\begin{aligned}
\iota_S &= \alpha_S - \frac{|S|\pi(S)}{d(p+dq)} \\
&= \left(\frac{1}{p} - \frac{q|S|}{p(p+dq)} \right) \pi(S) - \frac{|S|\pi(S)}{d(p+dq)} \\
&= \pi(S) \left(\frac{1}{p} - \frac{q|S|}{p(p+dq)} - \frac{|S|}{d(p+dq)} \right) \\
&\geq \pi(S) \left(\frac{1}{p} - \frac{qd}{p(p+dq)} - \frac{|S|}{d(p+dq)} \right) \\
&= \pi(S) \left(\frac{1}{p} \left(1 - \frac{qd}{p+dq} \right) - \frac{|S|}{d(p+dq)} \right) \\
&= \pi(S) \left(\frac{1}{p} \left(\frac{p}{p+dq} \right) - \frac{|S|}{d(p+dq)} \right) \\
&\geq \pi(S) \left(\frac{1}{p+dq} - \frac{d}{d(p+dq)} \right) \\
&= 0.
\end{aligned}$$

■

We are now prepared to prove Theorem 12. Our proof CLOSELY follows the proof of Theorem 11.

Proof [Theorem 12] We first reduce the empirical KernelSHAP values to the distribution KernelSHAP values by using the error term (Definition 25). We have

$$\begin{aligned}
\mathbb{E}_{x \sim \mu^*} |\mathcal{K}_i(\mu^*, f, x)| &\leq \eta(X^*, \mu^*, f) + \frac{1}{n} \sum_{x^{(j)} \in X^*} |\mathcal{K}_i(X^*, f, x^{(j)})| \\
&\leq \eta(X^*, \mu^*, f) + \epsilon,
\end{aligned}$$

with the latter holding from the Theorem statement.

We now use the same argument we did for proving Theorem 11, the only difference is that the operators we use to express the KernelSHAP operator differ from the ones used for the true SHAP values. Nevertheless, we will see that the same properties hold. In particular, Lemma 30 implies that

$$\begin{aligned}
\mathcal{K}_i^* &= \sum_{S \subseteq [d]} \iota_S v_S^* \\
&= \sum_{S \subseteq [d] \setminus \{i\}} \iota_{S \cup \{i\}} v_{S \cup \{i\}}^* - \sum_{S \subseteq [d] \setminus \{i\}} \iota_S v_S^* \\
&= C_i^* - D_i^*.
\end{aligned}$$

The key idea is that C_i^* and D_i^* fulfill *precisely* the same qualities that A_i^* and B_i^* did. That is,

1. $C_i^*(F_f^*) \subseteq F_f^*$,

2. $C_i^*(W) \subseteq W$,
3. $D_i^*(F_f^*) \subseteq W$,
4. $(C_i^*)^{-1}(\{0\}) = \{0\}$,

where F_f^* and W are just as in the proof of Theorem 11. To see this, simply observe that C_i^* and D_i^* are linear combinations of the *same value operators* as A_i^* and B_i^* . Thus the arguments used to prove the properties of A_i^* and B_i^* (i.e. Lemmas 10, 18 and 41) equally apply as the only facts we *ever* used about the coefficients of these linear combinations was that the coefficients in the expression for A_i^* were nonnegative and also equal to $\frac{1}{d}$ in the case of $v_{[d]}^*$. Thus, the proof to this Theorem immediately follows by simply replacing A_i^* and B_i^* from the proof of Theorem 11 with C_i^* and D_i^* . ■

Appendix C. Proofs from Section 5

C.1. Proof of Lemma 13

Proof [Lemma 13] Let $f \in F_T$ be a T -determined function. By definition,

$$(v_S f)(x) = \mathbb{E}_{X \sim \mu} f(x_S, X_{S^c}) = \mathbb{E}_{X \sim \mu} f(x_{S \cap T}, x_{S \setminus T}, X_{T \setminus S}, X_{(S \cup T)^c}). \quad (6)$$

To prove Property 1 of the lemma, let $x, x' \in \text{supp}(\mu^*)$ satisfy $x_{S \cap T} = x'_{S \cap T}$. Because f is T -determined, changing coordinates of an input point outside T does not effect its function value. Thus changing $x_{S \setminus T}$ to $x'_{S \setminus T}$ in Equation 6 and noting $x_{S \cap T} = x'_{S \cap T}$ gives us

$$\begin{aligned} (v_S f)(x) &= \mathbb{E}_{X \sim \mu} f(x_{S \cap T}, x_{S \setminus T}, X_{T \setminus S}, X_{(S \cup T)^c}) \\ &= \mathbb{E}_{X \sim \mu} f(x'_{S \cap T}, x'_{S \setminus T}, X_{T \setminus S}, X_{(S \cup T)^c}) = (v_S f)(x'). \end{aligned}$$

Since x, x' were arbitrary, this implies $v_S f$ is $S \cap T$ -determined. To prove Part 2, we further simplify Equation 6 by noting that $T \subseteq S$ to get

$$(v_S f)(x) = \mathbb{E}_{X \sim \mu} f(x_T, x_{S \setminus T}, X_{S^c}).$$

Because $f \in F_T$, for any choice of X we have $f(x_T, x_{S \setminus T}, X_{S^c}) = f(x_T, x_{S \setminus T}, x_{S^c}) = f(x)$. Substituting this into the expectation above implies $(v_S f)(x) = f(x)$, completing the proof. ■

C.2. Proof of Lemma 15

We begin with a useful technical lemma that will help us connect pure operators (Definition 16) to derived Lie sub-algebras of \mathfrak{g}_Φ .

Lemma 31 *Let W be a set of linear operators that is closed under derivations. Then*

$$\text{span}(W) = \left\{ \sum_{i=1}^n \lambda_i w_i : w_1, \dots, w_n \in W, \lambda_1, \dots, \lambda_n \in \mathbb{C}, n \in \mathbb{N} \right\},$$

is a Lie algebra.

Proof [Lemma 31] It suffices to show that $\text{span}(W)$ is closed under derivations as it is by definition a vector space. To do so, we appeal to the linearity of derivations. Observe that for any $a, b, c \in W$ and $\lambda \in \mathbb{C}$ it holds that

1. $[a + b, c] = (a + b)c - c(a + b) = (ac - ca) + (bc - cb) = [a, c] + [b, c].$
2. $[a, b + c] = a(b + c) - (b + c)a = (ab - ba) + (ac - ca) = [a, b] + [a, c].$
3. $[a, \lambda b] = a(\lambda b) - \lambda(ba) = \lambda[a, b].$
4. $[\lambda a, b] = (\lambda a)b - b(\lambda a) = \lambda[a, b].$

Applying these properties, we see that for $\sum \lambda_i w_i, \sum \lambda'_i w'_i \in \text{span}(W)$,

$$\left[\sum_{i=1}^n \lambda_i w_i, \sum_{i=1}^{n'} \lambda'_i w'_i \right] = \sum_{i=1}^n \sum_{j=1}^{n'} \lambda_i \lambda'_j [w_i, w'_j].$$

The latter sum is a linear combination of elements from W as $[w_i, w'_j] \in W$ for all i, j . It follows that $\left[\sum_{i=1}^n \lambda_i w_i, \sum_{i=1}^{n'} \lambda'_i w'_i \right] \in \text{span}(W)$ as desired. \blacksquare

We first show that pure operators (Definition 16) are sufficient for constructing linear algebraic bases of all Lie sub-algebras $\mathfrak{g}_\Phi^{(i)}$ in the derived series of \mathfrak{g}_Φ .

Lemma 32 *Let $V^{(0)} = V$ be the set of all pure operators. For $i \geq 1$, let $V^{(i)} = \{[v, w] : v, w \in V^{(i-1)}\}$. Then for all $i \geq 0$, $V^{(i)}$ is closed under derivations and satisfies $\text{span}(V^{(i)}) = \mathfrak{g}_\Phi^{(i)}$.*

Proof [Lemma 32] We proceed by induction on i . For the base case, $V^{(0)}$ is by definition the minimal set closed under derivations that contains $\{v_S : S \subseteq [d]\}$. Thus Lemma 31 implies that $\text{span}(V^{(0)})$ is a Lie algebra containing $\{v_S : S \subseteq [d]\}$ which implies that $\mathfrak{g}_\Phi^{(0)} \subseteq \text{span}(V^{(0)})$. On the other hand $V^{(0)}$ is a clear subset of $\mathfrak{g}_\Phi^{(0)}$ as $V^{(0)}$ is the minimal set containing $\{v_S : S \subseteq [d]\}$ that is closed under derivations. Thus, since $\mathfrak{g}_\Phi^{(0)}$ is a Lie algebra and thus a vector space, we have $\text{span}(V^{(0)}) \subseteq \mathfrak{g}_\Phi^{(0)}$ which implies equality.

Next, suppose the inductive hypothesis holds for $i - 1$. We first show that $V^{(i)}$ is closed under derivations. Let $[v, w]$ and $[v', w']$ be two elements in $V^{(i)}$ with $v, w, v', w' \in V^{(i-1)}$. Since $V^{(i-1)}$ is closed under derivations (inductive hypothesis), it follows that $[v, w]$ and $[v', w']$ are themselves elements of $V^{(i-1)}$. The definition of $V^{(i)}$ implies their derivation $[[v, w], [v', w']]$ is an element of $V^{(i)}$, which proves closure.

Next, by the definition of $V^{(i)}$, we see that

$$\text{span}(V^{(i)}) \subseteq [\text{span}(V^{(i-1)}), \text{span}(V^{(i-1)})] = [\mathfrak{g}_\Phi^{(i-1)}, \mathfrak{g}_\Phi^{(i-1)}] = \mathfrak{g}_\Phi^{(i)}.$$

In the other direction, Lemma 31 implies that $\text{span}(V^{(i)})$ is itself a Lie algebra (as $V^{(i)}$ is closed under derivations). Since $\mathfrak{g}_\Phi^{(i)}$ is defined as the smallest Lie algebra that contains $[v, w]$ for $v, w \in \mathfrak{g}_\Phi^{(i-1)}$, it suffices to show that $\text{span}(V^{(i)})$ contains this as well.

To this end let $v, w \in \mathfrak{g}_\Phi^{(i-1)}$. Applying the inductive hypothesis, we express them in their basis from $V^{(i-1)}$ by setting $v = \sum_{j=1}^n \lambda_j v_j$ and $w = \sum_{k=1}^m \mu_k w_k$ for $v_1, \dots, v_n, w_1, \dots, w_m \in V^{(i-1)}$, we see that

$$[v, w] = \sum_j \sum_k \lambda_j \mu_k [v_j, w_k].$$

This is clearly in $\text{span}(V^{(i)})$ as each element $[v_j, w_k]$ is in $V^{(i)}$ by definition. This completes the proof. \blacksquare

Next, we show how the sets $V^{(i)}$ interact with the subsets constructed in Lemma 17.

Lemma 33 (Size of Associated Subsets) $\alpha(v)$ satisfies the following two properties.

1. If $\alpha(v_1) = \alpha(v_2)$, then $[v_1, v_2] = 0$.
2. For all $v \in V^{(i)}$, $|\alpha(v)| \leq \max(d - i, 0)$.

Proof [Lemma 33] We begin with the first claim. Observe that for any $f \in F$, Lemma 17 implies $v_2 f \in F_{\alpha(v_2)}$. However, it also implies $v_1(F_{\alpha(v_1)}) = v_1(F_{\alpha(v_2)}) = 0$. Thus $v_1 v_2 f = 0$. Similarly $v_2 v_1 f = 0$. It follows that $[v_1, v_2]f = 0$ implying $[v_1, v_2] = 0$.

We prove the second claim by induction on i . The base case holds because all subsets have size at most d . Next, suppose the inductive hypothesis holds for $(i - 1)$.

Let $v \in V^{(i)}$. By definition, there exist $v_1, v_2 \in V^{(i-1)}$ for which $v = [v_1, v_2]$. Lemma 17 states that $\alpha(v) \subseteq \alpha(v_1) \cap \alpha(v_2)$. This gives us two cases.

First, if $\alpha(v_1) \neq \alpha(v_2)$, then

$$|\alpha(v)| \leq |\alpha(v_1) \cap \alpha(v_2)| < \max(|\alpha(v_1)|, |\alpha(v_2)|) \leq \max(d - i + 1, 0).$$

The strictness of this inequality implies that $|\alpha(v)| \leq \max(d - i, -1) = d - i$ which implies the inductive hypothesis holds for i .

Second, if $\alpha(v_1) = \alpha(v_2)$, then the first claim of the lemma implies $[v_1, v_2] = 0$ which immediately implies $\alpha([v_1, v_2]) \leq \max(d - i, 0)$, completing the inductive hypothesis. \blacksquare

We are finally prepared to prove Lemma 15.

Proof [Lemma 15] By Lemma 33, $\alpha(v) = \emptyset$ for all $v \in V^{(d)}$. Next, let $v \in V^{(d+1)}$. By definition, $v = [v_1, v_2]$ for $v_1, v_2 \in V^{(d)}$. Since $\alpha(v_1) = \alpha(v_2)$, Lemma 33 implies $[v_1, v_2] = 0$ which means $v = 0$. Thus $V^{(d+1)} = 0$. Since $V^{(d+1)}$ spans $\mathfrak{g}_{\Phi}^{(d+1)}$ as a vector space (Lemma 32), it follows that $\mathfrak{g}_{\Phi}^{(d+1)} = 0$ which means \mathfrak{g}_{Φ} is solvable. \blacksquare

C.3. Proof of Lemma 17

We begin with a useful lemma that characterizes the intersection of determined function spaces.

Lemma 34 (intersection of determined spaces) For $S, T \subseteq [d]$, $F_S \cap F_T = F_{S \cap T}$.

Proof [Lemma 34] We first show that $F_{S \cap T} \subseteq F_S \cap F_T$. Let $f \in F_{S \cap T}$, and let x, x' satisfy $x_S = x'_S$. Then $x_{S \cap T} = x'_{S \cap T}$ as well which implies $f(x_S) = f(x'_S)$ by the definition of a determined function. Thus f is S -determined meaning $f \in F_S$. We can similarly show $f \in F_T$.

Next, we show $F_S \cap F_T \subseteq F_{S \cap T}$. Let $f \in F_S \cap F_T$ and let x, x' satisfy $x_{S \cap T} = x'_{S \cap T}$. Using $f \in F_S$, $f \in F_T$, and $x_{S \cap T} = x'_{S \cap T}$, we get

$$\begin{aligned} f(x) &= f(x_{S \cap T}, x_{S \setminus T}, x_{T \setminus S}, x_{S^c \cap T^c}) \\ &= f(x_{S \cap T}, x_{S \setminus T}, x'_{T \setminus S}, x'_{S^c \cap T^c}) & (f \in F_S) \\ &= f(x_{S \cap T}, x'_{S \setminus T}, x'_{T \setminus S}, x'_{S^c \cap T^c}) & (f \in F_T) \\ &= f(x'_{S \cap T}, x'_{S \setminus T}, x'_{T \setminus S}, x'_{S^c \cap T^c}) = f(x'), & (x_{S \cap T} = x'_{S \cap T}) \end{aligned}$$

which implies the result. ■

We now prove Lemma 17.

Proof [Lemma 17] We first modify the extend of the criteria for $\alpha(v)$ to also apply to operators in $V \setminus \mathfrak{g}_\Phi^{(1)}$. We say that an operator $v \in V$ has a *nice* subset S if

1. $v(F_T) \subseteq F_{S \cap T}$ for all $T \subseteq [d]$.
2. If $T \subseteq S$ and $f \in F_T$, then $vf = \begin{cases} f & v \notin \mathfrak{g}_\Phi^{(1)} \\ 0 & v \in \mathfrak{g}_\Phi^{(1)} \end{cases}$.

We begin by showing that the set of nice subsets is closed under intersection. Suppose that S, S' are nice with respect to operator v . Then applying Lemma 34, we have

1. $v(F_T) \subseteq (F_{S \cap T} \cap F_{S' \cap T}) = F_{S \cap S' \cap T}$.
2. If $T \subseteq S \cap S'$ then $T \subseteq S$. Since S is nice, we have for all $f \in F_T$, $vf = \begin{cases} f & v \notin \mathfrak{g}_\Phi^{(1)} \\ 0 & v \in \mathfrak{g}_\Phi^{(1)} \end{cases}$.

We now define $\alpha(v)$ as the intersection of all nice subsets that v has. By our previous observation, $\alpha(v)$ itself is nice with respect to v and thus satisfies the first two properties of Lemma 17.

To complete the proof, it suffices to show that $\alpha(v)$ is well defined for all $v \in V$, and that it also satisfies Property 3 of Lemma 17. To this end, let $V' \subseteq V$ denote the set of all operators in V that have at least one nice subset. We claim that V' is closed under derivations. To see this, let $v, w \in V'$.

For any $T \subseteq [d]$, observe that because $\alpha(v), \alpha(w)$ are nice w.r.t. v, w ,

$$(vw)(F_T) \subseteq v(F_{\alpha(w) \cap T}) \subseteq F_{\alpha(v) \cap \alpha(w) \cap T},$$

$$(wv)(F_T) \subseteq w(F_{\alpha(v) \cap T}) \subseteq F_{\alpha(w) \cap \alpha(v) \cap T}.$$

Since $F_{\alpha(v) \cap \alpha(w) \cap T}$ is a vector space, it follows that $[v, w](F_T) \subseteq F_{\alpha(v) \cap \alpha(w) \cap T}$ thus showing that $\alpha(v) \cap \alpha(w)$ satisfies Property 1 of being a nice subset.

Next, let $T \subseteq \alpha(v) \cap \alpha(w)$. Let $\lambda_v = 1$ ($v \notin \mathfrak{g}_\Phi^{(1)}$) and $\lambda_w = 1$ ($w \notin \mathfrak{g}_\Phi^{(1)}$). Applying Property 2 of nice subsets to $\alpha(v), \alpha(w)$, we have that for any $f \in F_T$,

$$[v, w]f = (vw - wv)f = v\lambda_w f - w\lambda_v f = \lambda_v \lambda_w f - \lambda_w \lambda_v f = 0.$$

Thus, $\alpha(v) \cap \alpha(w)$ is nice with respect to $[v, w]$. Moreover, by the definition of α , this implies that $\alpha([v, w]) \subseteq \alpha(v) \cap \alpha(w)$.

Having verified all three properties, all that is left to show is that $V' = V$. To do so, observe that V' contains v_S for all $S \subseteq [d]$ as S is clearly a nice subset with respect to v_S (Lemma 13). Thus V' is a set closed under derivations that contains $\{v_S : S \subseteq [d]\}$. The definition of V implies that $V \subseteq V'$, and this implies equality as desired. ■

Appendix D. Definitions and Theorems about Lie Algebras

Definition 35 (Solvable Lie Algebra) For any Lie algebra, \mathfrak{g} , define the following:

1. Its derivation $[\mathfrak{g}, \mathfrak{g}]$ is the Lie sub-algebra generated by $\{[v_1, v_2] : v_1, v_2 \in \mathfrak{g}\}$.
2. Its derived series is the sequence $\mathfrak{g}^{(i)} = [\mathfrak{g}^{(i-1)}, \mathfrak{g}^{(i-1)}]$ with $\mathfrak{g}^{(0)} = \mathfrak{g}$.

Finally, \mathfrak{g} is solvable if there exists n such that $\mathfrak{g}_{\Phi}^{(n)} = 0$.

Definition 36 (Representation of Lie Algebra) Let \mathfrak{g} be a Lie algebra and let $GL(W)$ denote the general linear algebra over some vector space W . A representation of \mathfrak{g} is a pair (ρ, W) where W is a vector space, and $\rho : \mathfrak{g} \rightarrow GL(W)$ maps each element of \mathfrak{g} to a linear transformation over W such that for all $v, w \in \mathfrak{g}$,

1. $\rho(av + bw) = a\rho(v) + b\rho(w)$ for all $a, b \in \mathbb{C}$.
2. $\rho([v, w]) = \rho(v)\rho(w) - \rho(w)\rho(v)$.

(ρ, W) is said to be finite dimensional if W is.

Theorem 37 (Lie's Theorem) Let \mathfrak{g} be a solvable Lie algebra, and (ρ, W) be a finite dimensional representation. Then there exists a basis of W under which $\rho(g)$ is upper triangular for all $g \in \mathfrak{g}$.

Appendix E. Constructing a Finite Dimensional Representation of \mathfrak{g}_{Φ}

Definition 38 (Extended Space of Value Operators) We let V^* denote the space of all operators that can be obtained through linear combinations and compositions of value operators. That is,

$$V^* = \text{span} \{v_1 v_2 \dots v_m : m \in \mathbb{N}, v_1, \dots, v_m \in \{v_S : S \subseteq [d]\}\}.$$

Definition 39 (Localized Subspace) For $f \in F$, define its localized subspace,

$$F_f = V^* f = \{v f : v \in V^*\},$$

as the set of all functions that can be obtained from f by applying an operator in V^* to it.

We will use F_f to construct a finite dimensional representation of the Shapley Lie algebra. The definition of a finite dimensional representation can be found in Definition 36 of Section D.

Lemma 40 (Local Representation of Shapley Lie Algebra) Let $f \in F$ be any function. For all $v \in \mathfrak{g}_{\Phi}$, let $\rho_f(v)$ be defined as the restriction of v to F_f . Then $f \in F_f$, and (ρ_f, F_f) is a well-defined finite dimensional representation of the Shapley Lie algebra \mathfrak{g}_{Φ} .

Proof [Lemma 40] The fact that $f \in F_f$ is immediate: $v_{[d]}$ is precisely the identity operator and thus $V^* f$ contains $v_{[d]} f = f$.

To prove (ρ_f, F_f) is a well defined representation, it suffices to show that $V^* f$ is a v -invariant subspace of F (meaning $v(V^* f) \subseteq V^* f$) for all $v \in \mathfrak{g}_{\Phi}$ (the rest of the properties follow from basic properties of linear transformations). Let \mathfrak{g}'_{Φ} denote the set of all $v \in \mathfrak{g}_{\Phi}$ such that $V^* f$ is v -invariant. \mathfrak{g}'_{Φ} clearly contains all value operators v_S as $v_S(v_1 \dots v_m f)$ is itself a composition of value operators applied to f . \mathfrak{g}'_{Φ} is also closed under linear combinations (because V^* is a span of all compositions) and derivations (because it is closed under matrix multiplication). Thus, \mathfrak{g}'_{Φ} forms a Lie algebra that contains all value operators, and thus \mathfrak{g}'_{Φ} must contain \mathfrak{g}_{Φ} (by the minimality of \mathfrak{g}_{Φ}). This proves (ρ_f, F_f) is well-defined.

Next, we show F_f is a finite dimensional vector space. The key idea is to show that if the element $v_1 v_2 \dots v_m f$ satisfies $v_1 \in \{v_2, \dots, v_m\}$, then $v_1 \dots v_m f = v_2 \dots v_m f$. In short, v_1 has no effect after it has already been applied in a product.

To prove this, let $v_i = v_{S_i}$ for $S_i \subseteq [d]$ and $v_1 \in \{v_2, \dots, v_m\}$. Then Property 1 of Lemma 13 implies

$$v_2 \dots v_m f = v_{S_2} \dots v_{S_m} f \in F_{S_2 \cap \dots \cap S_m}.$$

Applying Property 2 of Lemma 13 with $S_2 \cap \dots \cap S_m \subseteq S_1$ implies that v_{S_1} acts as the identity map, meaning that

$$v_1 (v_2 \dots v_m f) = v_{S_1} (v_2 \dots v_m f) = v_2 \dots v_m f.$$

Applying this claim repeatedly, it follows that any product $v_1 \dots v_m f$ can be reduced to one in which $m \leq 2^d$ as there are at most 2^d distinct subsets of $[d]$. It follows that

$$\begin{aligned} V^* &= \text{span} \{v_1 v_2 \dots v_m : m \in \mathbb{N}, v_1, \dots, v_m \in \{v_S : S \subseteq [d]\}\} \\ &= \text{span} \left\{ v_1 v_2 \dots v_m : m \in \mathbb{N}, v_1, \dots, v_m \in \{v_S : S \subseteq [d]\}, m \leq 2^d \right\}. \end{aligned}$$

Because the latter space is a span of a finite number of elements, it follows that $V^* f$ itself is finite which completes the proof. \blacksquare

Lemma 41 (Localized Subspace if preserved by A_i and B_i) *Let F_f be the localized subspace of f . Then $f \in F_f$ and $A_i(F_f), B_i(F_f) \subseteq F_f$, where A_i, B_i are as defined in Definition 9.*

Proof [Lemma 41] This immediately follows from Lemma 40 along with the fact that A_i and B_i are both in \mathfrak{g}_Φ seeing as they are linear combinations of value operators. \blacksquare

Appendix F. Finding Counterexamples with a Linear Program

The key observation that makes it possible to use a Linear Program to construct counterexamples such as the one in Figure 1 is that the value function and therefore also the SHAP values themselves are linear in the function values. This holds for both, the observational and interventional SHAP value function.

In order to exploit this linear nature of the SHAP values, we consider a piecewise constant function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, that only takes finitely many values. We achieve this by defining f on a two dimensional $d_1 \times d_2$ -grid, where d_1 and d_2 define the size of the grid for Features 1 and 2, respectively. Then we can represent f as a $(d_1 \cdot d_2)$ -dimensional vector and the SHAP values for Feature 1 can be computed via matrix multiplication $\Phi_1 f$ where Φ_1 is a $d_1 \cdot d_2 \times d_1 \cdot d_2$ -dimensional matrix.

Now as a constraint for the Linear Program we can simply set $\Phi_1 f = 0$ which would force all the SHAP values on the extended support to be zero. If we only want to restrict the SHAP values inside the support to be zero, we can simply set the rows in Φ_1 to zero, if they correspond to a grid cell, that lies outside the support.

Finally the objective of the Linear Program will be to find a function, that does depend on Feature 1. This can be achieved by choosing two entries in the vector f that correspond to two input points with the same x_2 -value but different x_1 -values. If they differ, the function depends on Feature 1. Therefore maximizing their difference gives the desired results. However, many different approaches are possible here.

Figure 1 above shows one of these counterexamples that can be found using this Linear Program. Further counterexamples for smaller grids are displayed in Figure 2 and Figure 3, where we set $d_1 = d_2 = 3$ and $d_1 = d_2 = 4$, respectively.

If we strengthen the constraint on the SHAP values to be zero on the full extended support, as described above, the Linear Program is not able to find a counterexample, which we would expect considering our theoretical results above. Figure 4 displays one final example where the SHAP values of Feature 1 are zero on the whole extended support and the function indeed does not depend on Feature 1.

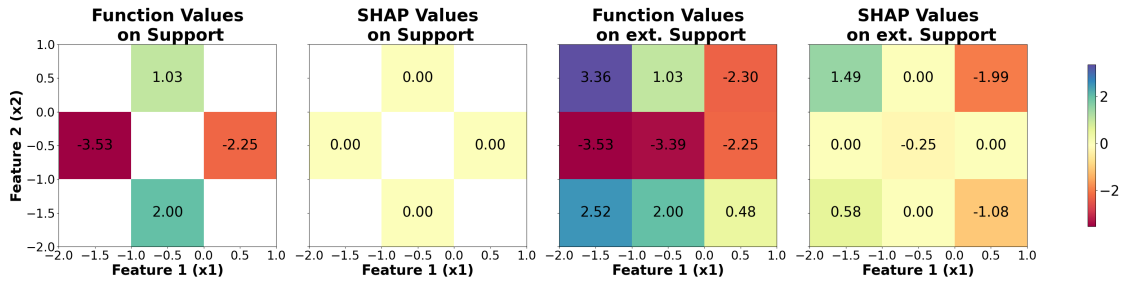


Figure 2: Example of a function where the aggregate SHAP value of Feature 1 is 0, yet the function depends on this feature on a 3×3 -grid. (a): Function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, supported on only 4 of the grid cells with the color depicting the function value. The function clearly depends on both Features 1 and 2. **(b):** Point-wise SHAP values $\phi_1(\mu, f, x)$ of Feature 1 are constantly 0 on the support. **(c) and (d):** Function and SHAP values on the extended support. Here the SHAP values are not constantly 0 any more.

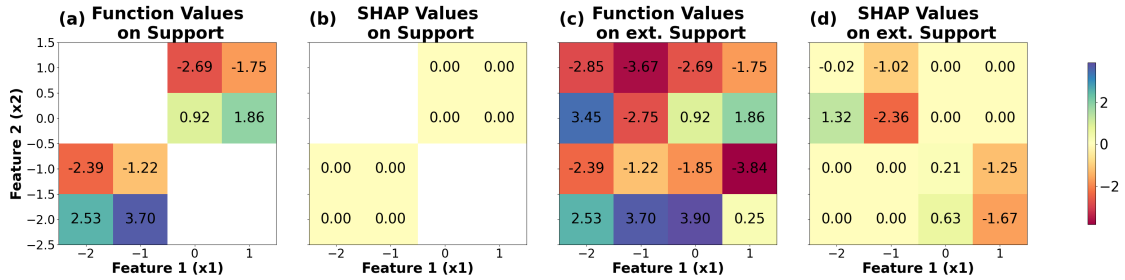


Figure 3: Example of a function where the aggregate SHAP value of Feature 1 is 0, yet the function depends on this feature on a 4×4 -grid. (a): Function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, supported on only 8 of the grid cells with the color depicting the function value. The function clearly depends on both Features 1 and 2. **(b):** Point-wise SHAP values $\phi_1(\mu, f, x)$ of Feature 1 are constantly 0 on the support. **(c) and (d):** Function and SHAP values on the extended support. Here the SHAP values are not constantly 0 any more.

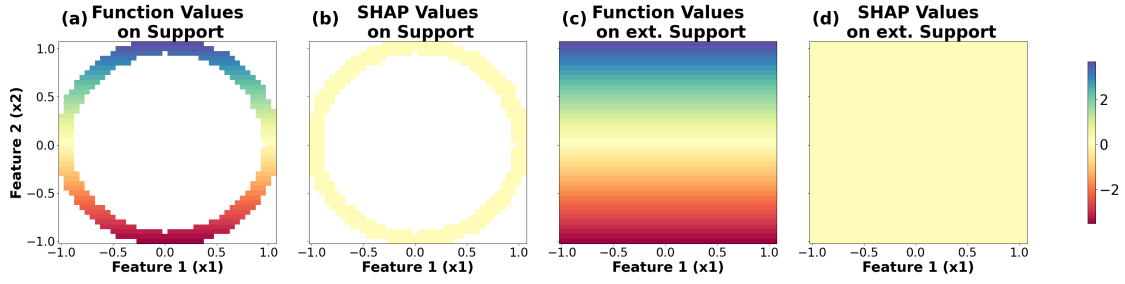


Figure 4: Example of a function where the aggregate SHAP value of Feature 1 is 0 on the whole extended support and the function does not depend on this feature. **(a):** Function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, supported on a ring with the color depicting the function value. The function solely depends on Feature 2. **(b):** Point-wise SHAP values $\phi_1(\mu, f, x)$ of Feature 1 are constantly 0 on the support. **(c) and (d):** Function and SHAP values on the extended support. Here the SHAP values are constantly 0 on the extended support as well.

Appendix G. Use cases of mean absolute SHAP in literature

This section gives an overview over different use cases of the mean absolute SHAP value in science literature. Table 1 holds an incomplete list of examples from recent years. In most applications the focus lies on the top features and while the possibility of doing feature selection based on the mean absolute SHAP value is often mentioned, e.g., by [Sharma Timilsina et al. \(2024\)](#), scientists are careful in actually applying it. It is merely used to select features for further analysis and interpretation.

Table 1: Collection of some exemplary use cases of the mean absolute SHAP value in literature.

Reference	Scientific field	Use of mean abs. SHAP
Greenwood et al. (2024)	Environmental Science	They investigate the influence of environmental and socioeconomic factors on the use of safely managed drinking water services. The features are grouped and the mean absolute SHAP value is calculated for each of the 5 groups.
Sharma Timilsina et al. (2024)	Physical Science	They use ML to predict the heating value of different types of waste and compute the mean absolute SHAP value to analyze the influence of the 8 features with a focus on the most important features.
Delavaux et al. (2023)	Environmental Science	They want to identify drivers of non-native plant invasions in native ecosystems. Mean absolute SHAP values are interpreted as feature importance.
Bernard et al. (2023)	Medical Science	They predict the physiological age based on biological values routinely assessed for diagnosis and treatment-monitoring. They use mean absolute SHAP values to identify the top-20 out of 48 variables.
Ekanayake et al. (2022)	Physical Science	They predict the compressive strength of concrete depending on its constituents and use the mean absolute SHAP values for interpretation of the 8 features. They focus on both, top and bottom features.

Table 1 – continued from previous page

Reference	Scientific field	Use of mean abs. SHAP
Wang et al. (2022)	Environmental Science	They want to understand pollutant removal in wastewater treatment plants and use the mean absolute SHAP value to choose the top-4 out of 32 features and take a deeper look into their influence.
Rane et al. (2022)	Medical Science	They analyze the IMAGEN data set to predict, based on brain images, whether a person is going to misuse alcohol. The features are considered most significant if they have mean abs SHAP value at least two times higher than the average mean abs SHAP value across all features.
Qiu et al. (2022)	Medical Science	They develop a deep learning framework to classify different causes for dementia and differentiate them from Alzheimer's disease. The mean absolute SHAP values are used for interpretation with a focus on the top-15 features.
Chen et al. (2022)	Physical Science	They classify proteins into self-assembling and partner-dependent proteins. The mean absolute SHAP values are used to interpret the influence of the features. All features are considered for the analysis with a focus on the top features.
Yang et al. (2022)	Physical Science	They design a machine learning implementation for the discovery of innovative polymers with ideal performance. The top-12 important molecular descriptors are identified using aggregate SHAP values.