Algorithms to Estimate Shapley Value Feature Attributions

Hugh Chen

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2022

Reading Committee:
Su-In Lee, Chair
Tim Althoff
Linda Shapiro

Program Authorized to Offer Degree:
Paul G. Allen School of Computer Science and Engineering

University of Washington

Abstract

Algorithms to Estimate Shapley Value Feature Attributions

Hugh Chen

Chair of the Supervisory Committee:
Paul G. Allen Professor Su-In Lee
Paul G. Allen School of Computer Science and Engineering

Black box machine learning models are increasingly prevalent. Their complex nature enables strong predictive accuracy but also makes them hard for humans to understand. One popular strategy to bridge the gap between complex models and interpretable models is to explain complex models using local feature attributions where a single sample's prediction is attributed to each of its features. In this class of explanation methods, Shapley value feature attributions have recently caught on. Although the Shapley value is appealing for its nice properties, it is NP-hard to compute in general. Here, we describe several works centered around tractably estimating the two most common variants of Shapley value feature attributions: marginal and conditional Shapley values.

TABLE OF CONTENTS

GLOSSARY

FEATURES: features within the dataset, of which there are d.

MODEL: machine learning model $f : \mathbb{R}^d \to \mathbb{R}$.

EXPLICAND: sample being explained $x^e \in \mathbb{R}^d$.

BASELINE: sample(s) being compared to $x^b \in \mathbb{R}^d$.

LOCAL FEATURE ATTRIBUTION: importance of each feature for a given explicand's prediction $\phi(f, x^e) \in \mathbb{R}^d$.

PLAYERS: a set of participants in a coalitional game $D = \{p_1, \cdots, p_d\}$.

COALITIONAL GAME: a game, also known as a cooperative game, which is defined by a set (value) function which maps from subsets of players to a value $v : 2^D \to \mathbb{R}$.

THE SHAPLEY VALUE: a unique solution concept in coalitional game theory to allocate credit to players in a coalitional game $\phi(v) \in \mathbb{R}^d$.

DEDICATION

to my fiance, Rachel; my parents, Hsinchun and Hsiao-Hui; my sister, Hillary; my dog, Tubulin; my grandparents; and all my friends and family ☺

Chapter 1

INTRODUCTION

Machine learning models are increasingly prevalent because they have matched or surpassed human performance in many applications: these include Go [176], poker [138], Starcraft [201], protein folding [92], language translation [88], and more. One critical component in their success is flexibility, or expressive power [22, 112, 35], which has been facilitated by more complex models and improved hardware [182]. Unfortunately, their flexibility also makes models opaque, or challenging for humans to understand. Combined with the tendency of machine learning to rely on shortcuts [69] (i.e., unintended learning strategies that fail to generalize to unseen data), there is a growing demand for model interpretability [52]. This demand is reflected in increasing calls for explanations by diverse regulatory bodies, such as the General Data Protection Regulation's "right to explanation" [171] and the Equal Credit Opportunity Act's adverse action notices [102].

There are many possible ways to explain machine learning models (e.g., counterfactuals, exemplars, surrogate models, etc.), but one extremely popular approach is *local feature attribution*. In this approach, individual predictions are explained by an attribution vector $\phi \in \mathbb{R}^d$, with d being the number of features used by the model. One prominent example is LIME [160], which fits a simple interpretable model that captures the model's behavior in the neighborhood of a single sample; when a linear model is used, the coefficients serve as attribution scores for each feature. In addition to LIME, many other methods exist to compute local feature attributions [160, 121, 123, 175, 20, 47, 189]. One popular class of approaches is *additive feature attribution methods*, which are those whose attributions sum to a specific value, such as the model's prediction [121].

To unify the class of additive feature attribution methods, Lundberg and Lee [121] in-

troduced SHAP as a unique solution determined by additional desirable properties (Section 2.3). Its uniqueness depends on defining a coalitional game (or set function) based on the model being explained (a connection first introduced in Strumbelj and Kononenko [183]). Lundberg and Lee [121] initially defined the game as the expectation of the model's output when conditioned on a set of observed features. However, given the difficulty of computing conditional expectations in practice, the authors suggested using a marginal expectation that ignores dependencies between the observed and unobserved features. This point of complexity has led to distinct Shapley value approaches that differ in how they remove features [108, 188, 87, 80, 42], as well as subsequent interpretations of how these two approaches relate to causal interventions [87, 80] or information theory [34, 42]. Moving forward, we will refer to all feature attributions based on the Shapley value as *Shapley value explanations*.

Alongside the definition of the coalitional game, another challenge for Shapley value explanations is that calculating them has computational complexity that is exponential in the number of features. The original SHAP paper [121] therefore discussed several strategies for approximating Shapley values, including weighted linear regression (KernelSHAP [121]), sampling feature combinations (IME [183]), and several model-specific approximations (LinearSHAP [121, 29], MaxSHAP [121], DeepSHAP [121, 30]). Since the original work, other methods have been developed to estimate Shapley value explanations more efficiently, using model-agnostic strategies (permutation [25], multilinear extension [146], FastSHAP [90]) and model-specific strategies (linear models [29], tree models [123], deep models [30, 9, 203]).

The abundance of distinct algorithms to estimate Shapley value explanations coupled with their complexity have made this literature inaccessible, which is problematic given the widespread use of these approaches. We describe twenty-four such algorithms by disentangling their complexity into two factors: (1) the approach to remove features and (2) the tractable estimation strategy [32][1] (Chapter 2). These factors provide a natural lens through which we can better comprehend and compare Shapley value explanation algorithms. Un-

[1]This article is currently under review at Nature Machine Intelligence.

derstanding these factors can also ensure the algorithms are not misused and help identify fundamental limitations and important future research directions. In addition, this chapter will also serve to carefully introduce concepts of feature attributions, the Shapley value, and Shapley value explanations.

Then, we will describe earlier projects that specifically aim to produce Shapley value explanations tractable for key model types: linear, tree, deep, and model pipelines.

First, we develop algorithms to estimate Shapley value explanations for linear models where they are easier to compute and compare in order to intuitively understand **tradeoffs between marginal and conditional Shapley values** [29] (Chapter 3). We find that conditional Shapley values spread credit according to statistical dependencies in the data and marginal Shapley values provide a closer explanation of the model's functional form.

Next, we develop methods to estimate **Shapley value feature attributions for tree models** [123][2], where we find that it is surprisingly possible to tractably estimate marginal Shapley values exactly and conditional Shapley values approximately (Chapter 4).

Then, we validate a **self-supervised learning approach applied to physiological signals** [31] (Chapter 5). In this application, we train tree models based on features extracted from neural networks. Using the tractable algorithms we developed to explain tree models, we explain adverse outcomes in a surgical setting in terms of the physiological signals that contributed to risk.

We develop an approach to estimate **Shapley value feature attributions for series of models** [30] (Chapter 6). This approach generalizes the previous application where a tree model sits on top of a neural network (i.e., a series of models). By propagating the explanations through each model, we are able to explain a series of models in terms of the original input space.

---

[2]My primary contribution to this paper was the development of the tractable algorithm to estimate marginal Shapley values exactly for tree models.

Chapter 2

SHAPLEY VALUE EXPLANATION ALGORITHMS

2.1 Introduction

Machine learning models are increasingly prevalent because they have matched or surpassed human performance in many applications: these include Go [176], poker [138], Starcraft [201], protein folding [92], language translation [88], and more. One critical component in their success is flexibility, or expressive power [22, 112, 35], which has been facilitated by more complex models and improved hardware [182]. Unfortunately, their flexibility also makes models opaque, or challenging for humans to understand. Combined with the tendency of machine learning to rely on shortcuts [69] (i.e., unintended learning strategies that fail to generalize to unseen data), there is a growing demand for model interpretability [52]. This demand is reflected in increasing calls for explanations by diverse regulatory bodies, such as the General Data Protection Regulation's "right to explanation" [171] and the Equal Credit Opportunity Act's adverse action notices [102].

There are many possible ways to explain machine learning models (e.g., counterfactuals, exemplars, surrogate models, etc.), but one extremely popular approach is *local feature attribution*. In this approach, individual predictions are explained by an attribution vector $\phi \in \mathbb{R}^d$, with d being the number of features used by the model. One prominent example is LIME [160], which fits a simple interpretable model that captures the model's behavior in the neighborhood of a single sample; when a linear model is used, the coefficients serve as attribution scores for each feature. In addition to LIME, many other methods exist to compute local feature attributions [160, 121, 123, 175, 20, 47, 189]. One popular class of approaches is *additive feature attribution methods*, which are those whose attributions sum to a specific value, such as the model's prediction [121].

To unify the class of additive feature attribution methods, Lundberg and Lee [121] introduced SHAP as a unique solution determined by additional desirable properties (Section 2.3). Its uniqueness depends on defining a coalitional game (or set function) based on the model being explained (a connection first introduced in [183]). Lundberg and Lee [121] initially defined the game as the expectation of the model's output when conditioned on a set of observed features. However, given the difficulty of computing conditional expectations in practice, the authors suggested using a marginal expectation that ignores dependencies between the observed and unobserved features. This point of complexity has led to distinct Shapley value approaches that differ in how they remove features [108, 188, 87, 80, 42], as well as subsequent interpretations of how these two approaches relate to causal interventions [87, 80] or information theory [34, 42]. Moving forward, we will refer to all feature attributions based on the Shapley value as *Shapley value explanations*.

Alongside the definition of the coalitional game, another challenge for Shapley value explanations is that calculating them has computational complexity that is exponential in the number of features. The original SHAP paper [121] therefore discussed several strategies for approximating Shapley values, including weighted linear regression (KernelSHAP [121]), sampling feature combinations (IME [183]), and several model-specific approximations (LinearSHAP [121, 29], MaxSHAP [121], DeepSHAP [121, 30]). Since the original work, other methods have been developed to estimate Shapley value explanations more efficiently, using model-agnostic strategies (permutation [25], multilinear extension [146], FastSHAP [90]) and model-specific strategies (linear models [29], tree models [123], deep models [30, 9, 203]). Of these two categories, model-agnostic approaches are more flexible but stochastic, whereas model-specific approaches are significantly faster to calculate. To better understand the model-agnostic approaches, we present a categorization of the approximation algorithms based on equivalent mathematical definitions of the Shapley value, and we empirically compare their convergence properties (Section 2.4). Then, to better understand the model-specific approaches, we highlight the key assumptions underlying each approach (Section 2.5).

These two sources of complexity, properly removing features and accurately approximat-

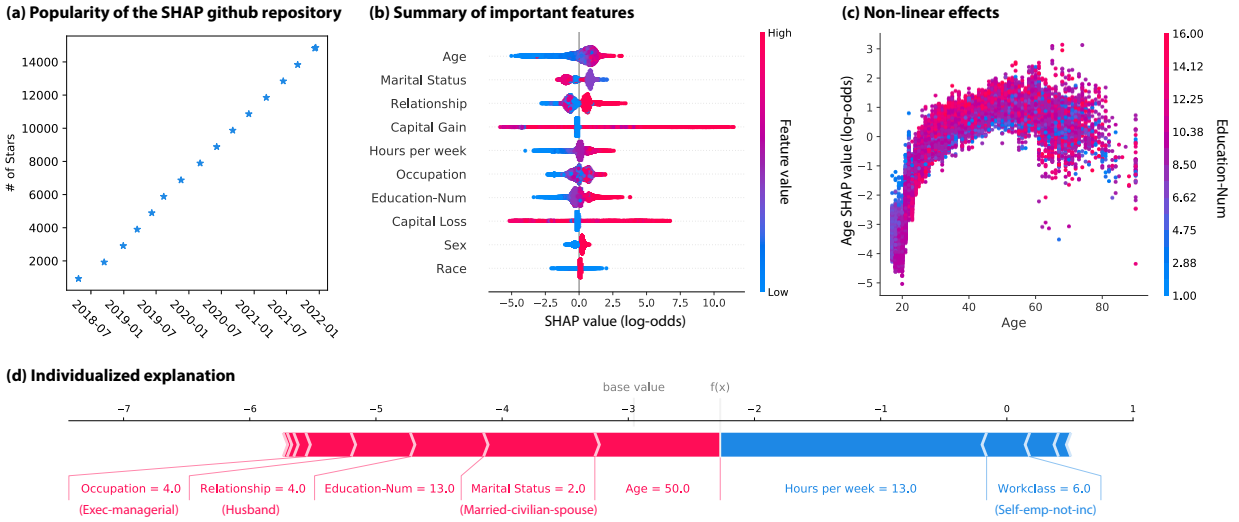


Figure 2.1: Shapley value explanations are popular and practical. (a) The large number of Github stars on shap (https://github.com/slundberg/shap), the most famous package to estimate Shapley value explanations, indicates their popularity. (b)-(d) A real-world example of Shapley value explanations for a tree ensemble model trained to predict whether individuals have income greater than 50,000 dollars based on census data. (b) Local feature attributions enable a global understanding of important features. (c) Local feature attributions help explain non-linear and interaction effects. (d) Local feature attributions explain how an individual's features influence their outcome.

ing Shapley values, have led to a wide variety of papers and algorithms on the subject. Unfortunately, this abundance of algorithms coupled with the inherent complexity of the topic have made the literature difficult to navigate, which can lead to misuse, especially given the popularity of Shapley value explanations (Figure 2.1a). To address this, we provide an approachable explanation of the sources of complexity underlying the computation of Shapley value explanations.

We discuss these difficulties in detail, beginning by introducing the preliminary concepts of feature attribution (Section 2.2) and the Shapley value (Section 2.3). Based on the various feature removal approaches, we then describe popular variants of Shapley value explanations as well as approaches to estimate the corresponding coalitional games (Section 2.4). Next, based on the estimation strategies, we describe model-agnostic and model-specific algorithms that rely on approximations and/or assumptions to tractably estimate Shapley value explana-

tions (section 2.5). These two sources of complexity provide a natural lens through which we present what is, to our knowledge, the first comprehensive survey of 24 distinct algorithms[1] that combine different feature removal and tractable estimation strategies to compute Shapley value explanations. Finally, we identify gaps and important future directions in this area of research throughout the article.

2.2 Feature attributions

Given a model f and features $x_1, \ldots, x_d$, feature attributions explain predictions by assigning scalar values that represent each feature's importance. For an intuitive description of feature attributions, we first consider linear models. Linear models of the form $f(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_d x_d$ are often considered interpretable because each feature is linearly related to the prediction via a single parameter. In this case, a common *global feature attribution* that describes the model's overall dependence on feature i is the corresponding coefficient β_i. For linear models, each coefficient β_i describes the influence that variations in feature x_i have on the model output.

Alternatively, it may be preferable to give an individualized explanation that is not for the model as a whole, but rather for the prediction $f(x^e)$ given a specific sample x^e. These types of explanations are known as *local feature attributions*, and the sample being explained (x^e) is called the *explicand*. For linear models, one reasonable local feature attribution is $\phi_i(f, x^e) = \beta_i x_i^e$, because it is exactly the contribution that feature i makes to the model's prediction for the given explicand. However, note that this attribution hides within it an implicit assumption that we want to compare against an alternative feature value of $x_i = 0$, but we may wish to account for other plausible alternative values, or more generally for the feature's distribution or statistical relationships with other features (Section 2.4).

Linear models offer a simple case where we can understand each feature's role via the model parameters, but this approach does not extend naturally to more complex model types.

[1] This count excludes minor variations of these algorithms.

For model types that are most widely used today, including tree ensembles and deep learning models, their large number of operations prevents us from understanding each feature's role by examining the model parameters. These flexible, non-linear models can capture more patterns in data, but they require us to develop more sophisticated and generalizable notions of feature importance. Thus, many researchers have recently begun turning to Shapley value explanations to summarize important features (Figure 2.1b), surface non-linear effects (Figure 2.1c), and provide individualized explanations (Figure 2.1d) in an axiomatic manner (Figure 2.2b).

2.3 Shapley values

Shapley values are a tool from game theory [172] designed to allocate credit to players in coalitional games. The *players* are represented by a set $D = \{1, \ldots, d\}$, and the *coalitional game* is a function that maps from subsets of the players to a scalar value. A game is represented by a subset function $v(S) : \mathcal{P}(D) \mapsto \mathbb{R}$, where $\mathcal{P}(D)$ is the power set of D (representing all possible subsets of players) (Figure 2.2a).

To make these concepts more concrete, we can imagine a company that makes a profit $v(S)$ determined by the set of employees $S \subseteq D$ that choose to work that day. A natural question is how to compensate the employees for their contribution to the total profit. Assuming we know the profit for all subsets of employees, Shapley values assign credit to an individual i by calculating a weighted average of the profit increase when i works with group S versus when i does not work with group S (the marginal contribution). Averaging this difference over all possible subsets S to which i does not belong ($S \subseteq D \setminus \{i\}$), we arrive at the definition of the Shapley value:

$$\overbrace{\phi_i(v)}^{i\text{'s Shapley value}} = \sum_{S \subseteq D \setminus \{i\}} \underbrace{\frac{|S|!(|D| - |S| - 1)!}{|D|!}}_{S\text{'s weight}} \overbrace{(v(S \cup \{i\}) - v(S))}^{i\text{'s marginal contribution}} \tag{2.1}$$

Shapley values offer a compelling way to spread credit in coalitional games, and they have

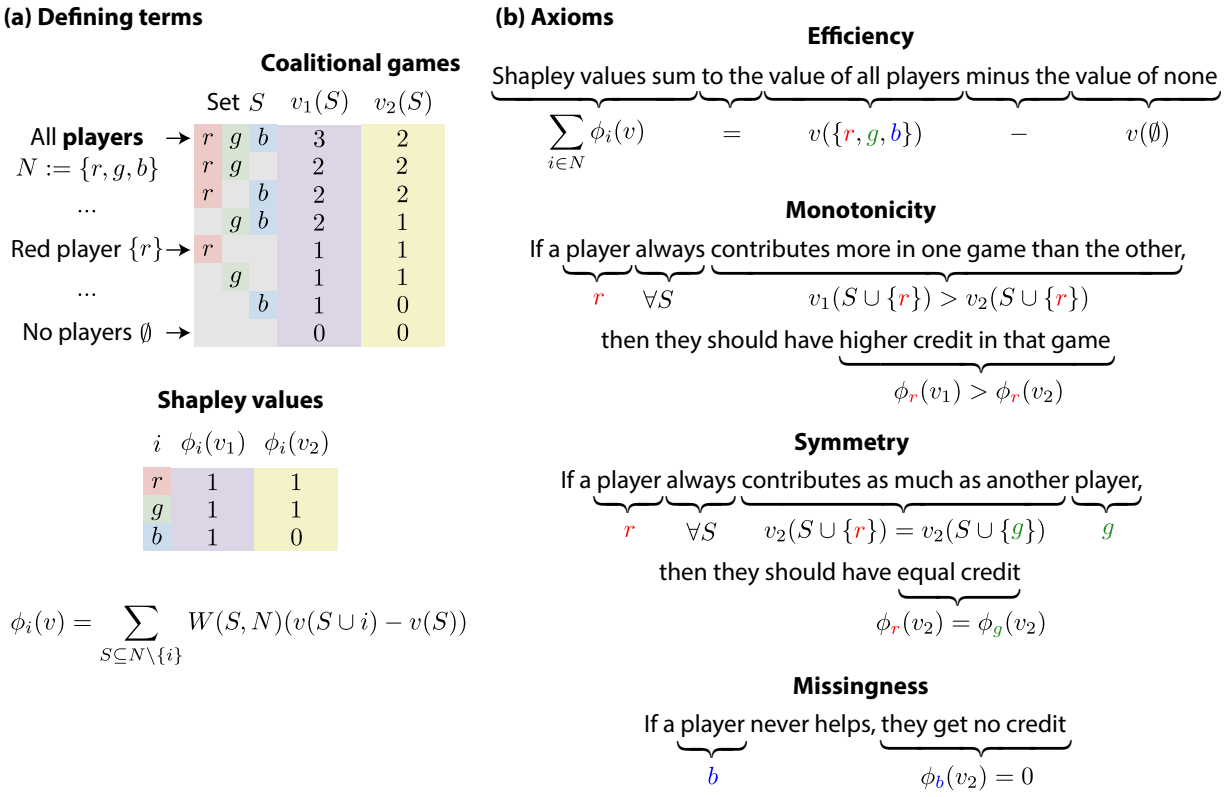


Figure 2.2: (a) Defining terms related to the Shapley value. Players either participate or abstain from the coalitional game, and the game maps from any subset of participating players to a scalar value. Shapley values are a solution concept to allocate credit to each player in a coalitional game. (b) A sufficient, but not exhaustive set of axioms that uniquely define the Shapley value.

been widely adopted in fields including computational biology [120, 140], finance [191, 192], and more [109, 12]. Furthermore, they are a unique solution to the credit allocation problem as defined by several desirable properties [172, 213] (Figure 2.2b).

2.4 Shapley value explanations

In this section, we present common strategies to define local feature attributions based on the Shapley value. We also present intuitive examples based on explaining linear models, and we discuss the tradeoffs between various approaches to removing features.

2.4.1 Machine learning models are not coalitional games

Although Shapley values are an attractive solution for allocating credit among players in coalitional games, our goal is to allocate credit among features $x_1, \ldots, x_d$ in a machine learning model $f(x) \in \mathbb{R}$. Machine learning models are not coalitional games by default, so to use Shapley values **we must first define a coalitional game $v(S)$ based on the model $f(x)$** (Figure 2.3a). The coalitional game can be chosen to represent various model behaviors, including the model's loss for a single sample or for the entire dataset [42], but our focus is the most common choice: explaining the prediction $f(x^e)$ for a single sample x^e.

When explaining a machine learning model, it is natural to view each feature x_i as a player in the coalitional game. However, we then must define what is meant by the presence or absence of each feature. Given our focus on a single explicand x^e, the presence of feature i will mean that the model is evaluated with the observed value x_i^e (Figure 2.3b). As for the absent features, we next consider how to remove them to properly assess the influence of the present features.

2.4.2 Removing features with baseline values

One straightforward way to remove a feature is to replace its value using a baseline sample x^b. That is, if a feature i is absent, we simply set that feature's value to be x_i^b. Then, the coalitional game is defined as $v(S) = f(\tau(x^e, x^b, S))$, where we define $\tau(x^e, x^b, S)_i = x_i^e$ if $i \in S$ or x_i^b otherwise (Figure 2.3c). In words, we evaluate the model on a new sample where present features are the explicand values and absent features are the baseline values. As shorthand notation, we will refer to $f(\tau(x^e, x^b, S))$ as $f(x_S^e, x_{\bar{S}}^b)$ in the remainder of the paper.

The Shapley values for this coalitional game are referred to as *baseline Shapley values* [188]. This approach is simple to implement, but the choice of the baseline is not straightforward and can be somewhat arbitrary. Many different baselines have been considered,

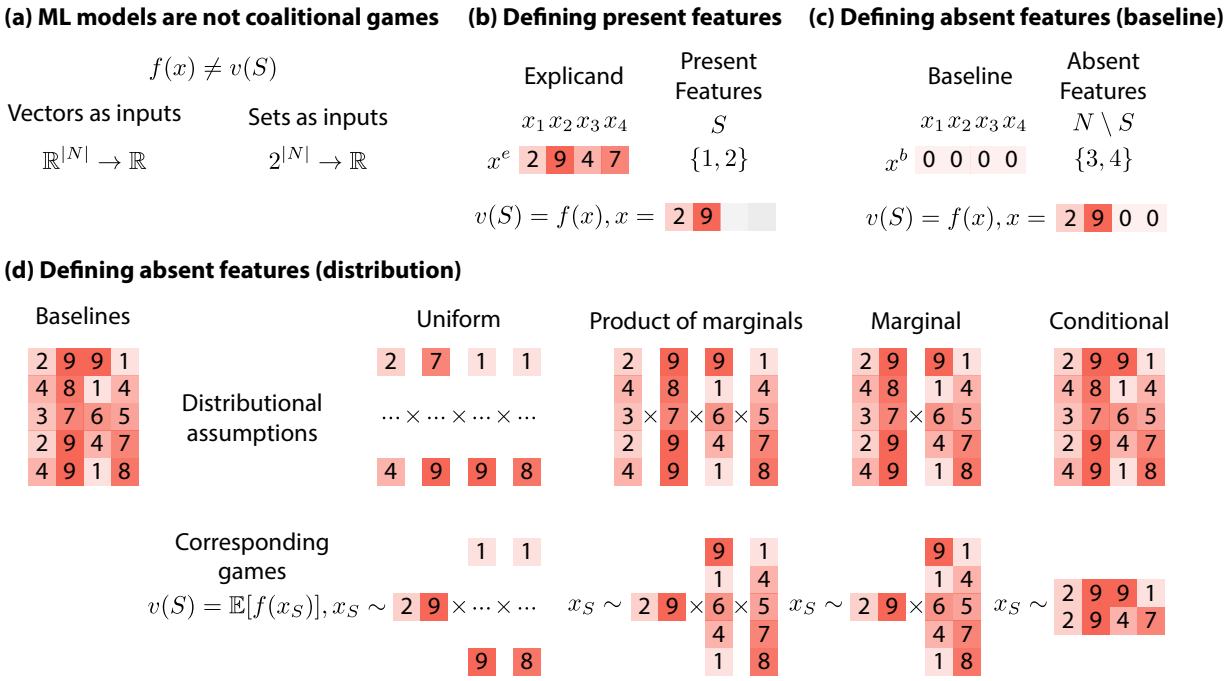


Figure 2.3: Empirical strategies for handling absent features. (a) Machine learning models have vector inputs and coalitional games have set inputs. For simplicity of notation we assume real-valued features, but Shapley value explanations can accommodate discrete features (unlike gradient-based methods). (b) Present features are replaced according to the explicand. (c) Absent features can be replaced according to a baseline. (d) Alternatively, absent features can be replaced according to a set of baselines with different distributional assumptions. In particular, the uniform approach uses the range of the baselines' absent features to define independent uniform distributions to draw absent features from. The product of marginals approach draws each absent feature independently according to the values seen in the baselines. The marginal approach draws groups of absent feature values that appeared in the baselines. Finally, the conditional approach only considers samples that exactly match on the present features. Note that this figure depicts empirically estimating each expectation; however, in practice, the conditional approach is estimated by fitting models (Section 2.5.1).

including an all-zeros baseline, an average across features[2], a baseline drawn from a uniform distribution, and more [189, 175, 63, 186, 95, 159]. Unfortunately, the choice of baseline heavily influences the feature attributions, and the criteria for choosing a baseline can be unclear. One possible motivation could be to find a neutral, uninformative baseline, but

[2]This baseline is most natural for image data [63, 186].

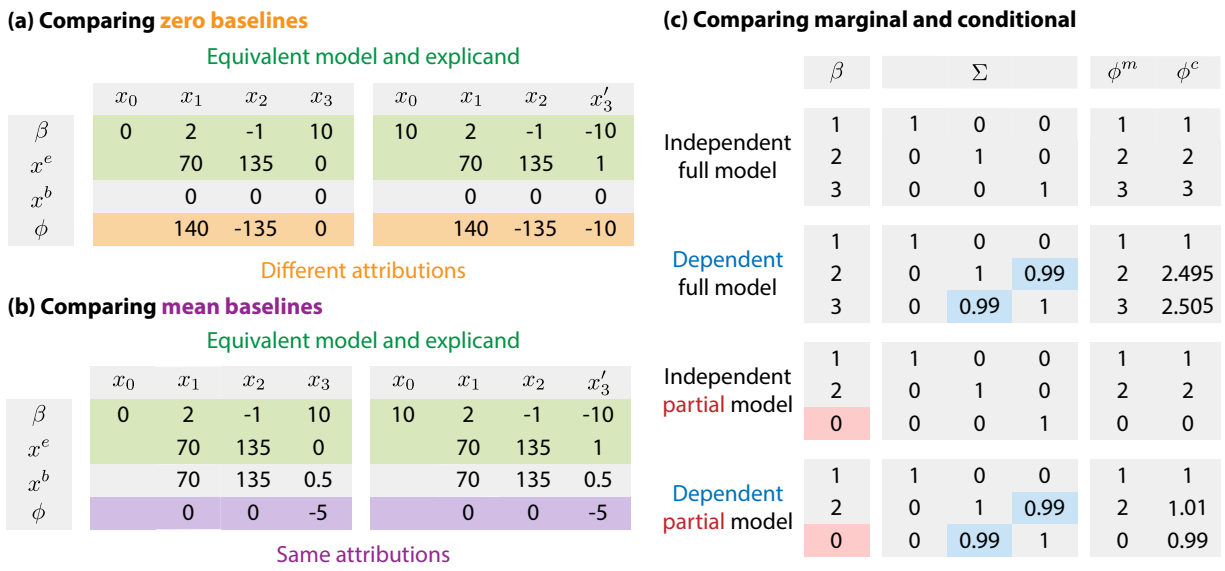


Figure 2.4: Shapley values for linear models. (a)-(b) A linear model (β), an explicand (x^e), a baseline (x^b), and baseline Shapley values (ϕ) where feature 1 represents height (inches), feature 2 represents weight (lbs), and feature 3 represents gender. Features x_3 and x_3' denote different ways to represent gender, where $x_3 = 1$ is male and $x_3' = 1$ is female. (a) The models and explicands on the left and right are equivalent, but a zero baseline has a different meaning in each example and thus produces different attributions. (b) In this case, we use an mean baseline, for which the encoding of gender does not affect the baseline Shapley values. (c) Comparing marginal and conditional Shapley values for different models and feature dependencies with explicand $x^e = (1, 1, 1)$ and baseline $x^b = (0, 0, 0)$. Vectors β (linear model coefficients), ϕ^m (marginal Shapley values), and ϕ^c (conditional Shapley values) have elements corresponding to x_1, x_2, x_3, and matrix Σ's columns and rows are x_1, x_2, x_3. The independent models have no correlation between features and the dependent models have a surrogate feature (a highly correlated pair of features). The full model has all non-zero coefficients whereas the partial model has a zero coefficient for the third feature.

such a baseline value may not exist. For these reasons, it is common to use a distribution of baselines instead of relying on a single baseline.

2.4.3 Removing features with distributional values

Rather than setting the removed features to fixed baseline values, another option is to average the model's prediction across randomly sampled replacement values; this may offer a better method to represent absent feature information. A first approach is to sample from the

conditional distribution for the removed features. That is, given an explicand x^e and subset $S \subseteq D$, we can consider the set of present features x^e_S and then sample replacement values for the absent features according to $x_{\bar{S}} \sim p(x_{\bar{S}} \mid x^e_S)$. In this case, the coalitional game is defined as the expectation of the prediction $f(x^e_S, x_{\bar{S}})$ across this distribution. There are several names for Shapley values with this coalitional game: conditional Shapley values [29], conditional expectation Shapley [188], and finally *conditional Shapley values* [80], which is how we will refer to them. Two issues with this approach are that estimating the conditional expectation is challenging (Section 2.5.1), and that the resulting explanations will spread credit among correlated features even if the model does not directly use all of them, which may not be desirable (Section 2.4.5).

An alternative approach is to use the *marginal distribution* when sampling replacement values. That is, we ignore the values for the observed features x^e_S and sample replacement values according to $x_{\bar{S}} \sim p(x_{\bar{S}})$. As in the previous case, the coalitional game is defined as the expectation of the prediction across this distribution. This approach is equivalent to averaging over baseline Shapley values with baselines drawn from the data distribution $p(x)$ [188]. It also has an interpretation based in causal interventions on the feature values, but not interventions on the real-world values the features represent, interventions on the feature values in the computer going into the machine learning model. This is equivalent to assuming a flat causal graph (i.e., a causal graph with no causal links among features) [87, 80]. The latter interpretation has led to the name marginal Shapley values, but to avoid ambiguity we opt for the name *marginal Shapley values* [80].

The conditional and marginal approaches are by far the most common feature removal approaches in practice. Two other formulations based on random sampling are (1) the uniform approach, where absent features are drawn from a uniform distribution covering the feature range, and (2) the product of marginals approach, where absent features are drawn from their individual marginal distributions (which assumes independence between all absent features) [47, 132]. However, these distributions make a strong assumption of independence between all features, which may be why marginal Shapley values, which make

a milder assumption of independence between the observed and unobserved features, are more commonly used. In addition, there are several other approaches for handling absent features in Shapley value-like explanations, but these can often be interpreted as approximations of the aforementioned approaches [42]. We visualized the three main removal approaches in Figure 2.3d, where, for simplicity, we show empirical versions that use a finite set of baselines (e.g., a training data set) to compute each expectation [188].

2.4.4 Shapley value explanations for linear models

To highlight intuitive differences between baseline, marginal, and conditional Shapley values, we consider the case of linear models where Shapley value explanations are easier to compute and compare (Figure 2.4). In the following examples, we consider different linear models, data distributions, and feature encodings to call attention to properties of these three types of Shapley value explanations.

Baseline Shapley values are simple and can be intuitive, but choosing an appropriate baseline is difficult. One common baseline is a sample with all zero feature values. However, we show that an all-zeros baseline can produce counterintuitive attribution values because the meaning of zero can be arbitrary (Figure 2.4a). In particular, we consider a case with equivalent models and explicand, but where the gender feature is encoded such that male is 0 (x_3) or male is 1 (x'_3). In the first case, the baseline Shapley value for x_3 is zero (Figure 2.4a left), signaling that being male does not impact the prediction; however, in the second case, the baseline Shapley value for x'_3 is -10 (Figure 2.4a right), suggesting that being male leads to lower predictions. These differing explanations are perhaps counterintuitive because the model and explicand are exactly equivalent, but the discrepancy arises because the meaning of zero is often arbitrary.

As an alternative to the all-zeros baseline, the mean baseline is arguably a reasonable choice for linear models. When using this baseline value, as in Figure 2.4b, the baseline Shapley value is zero for height and weight because the explicand's height and weight are equal to their average values. In addition, for a mean baseline, the baseline Shapley value is

the same regardless of how we encode gender. Although the mean baseline can work well for linear models (Section 2.5.2), it can be unappealing in other cases for two reasons. First, the mean of discrete features generally does not have a natural interpretation. For instance, the mean of the gender variable is half female and half male, a value that is never encountered in the dataset. This issue is compounded for non-ordinal discrete and categorical features with more than two possible values. Second, it may be impossible for any single baseline to represent the absence of feature information. For example, in images, removing features with a baseline cannot give credit to pixels that match the baseline [42, 186, 30]; for a mean baseline, this means that regions of images that resemble the mean will be biased towards lower importance.

Rather than baseline Shapley values, we may instead prefer to use marginal and conditional Shapley values, which we compare in Figure 2.4c. In this example, we compute Shapley value explanations for the same explicand and baseline, but with different models and data distributions (multivariate Gaussians with different covariances). We generate data in two ways: independent (zero covariance between features) or dependent (high covariance between features x_2 and x_3). In addition, we consider two models: full (all coefficients are non-zero) or partial (β_3 is zero).

Comparing the independent full model case to the dependent full model case, we can see that conditional Shapley values split credit between correlated features. This behavior may be desirable if we want to detect whether a model is relying on a protected class through correlated features. However, spreading credit can feel unnatural in the dependent partial model case, where the conditional Shapley value for feature x_3 (ϕ_3^c) is as high as the conditional Shapley value for feature x_2 (ϕ_2^c) even though feature x_3 is not explicitly used by the model ($\beta_3 = 0$). In particular, a common intuition is that features not algebraically used by the model should have zero attribution[3] [188]. One concrete example is within a

[3]This intuition is described by Sundararajan and Najmi [188] as the *dummy axiom* [188]. Notably, their axiom is defined relative to the model, whereas the game theory literature has an existing dummy axiom defined relative to the coalitional game [137]. Shapley value explanations always satisfy the original dummy axiom, as well as all other Shapley value axioms defined in terms of the coalitional game [172, 137].

mortality prediction setting (NHANES), where Chen et al. [29] [29] show that for a model that does not explicitly use body mass index (BMI) as a feature, conditional Shapley values still give high importance to BMI due to correlations with other influential features such as arm circumference and systolic blood pressure.

2.4.5 Tradeoffs between removal approaches

Given the many ways to formulate the coalitional game, or to handle absent features, a natural question is *which Shapley value explanation is preferred*? This question is frequently debated in Shapley value literature, with some papers defending marginal Shapley values [188, 87], others advocating for conditional Shapley values [9, 4, 42], and still others arguing for causal solutions [80]. Before discussing differences, one way the approaches are alike is that each Shapley value explanation variant always satisfies the same axioms for its corresponding coalitional game, although the interpretation of the axioms can differ; this point has been discussed in prior work [188], but it is important to avoid conflating axioms defined relative to the coalitional game and relative to the model. Below, we discuss tradeoffs between the two most popular approaches, marginal and conditional Shapley values, because these are most commonly implemented in public repositories and discussed in the literature.

As we have seen with linear models, conditional Shapley values tend to spread credit between correlated features, which can surface hidden dependencies [66], whereas marginal Shapley values yield attributions that are a description of the model's functional form [29]. This discrepancy arises from the distributional assumptions, where conditioning on a feature implicitly introduces information about all correlated features, thereby leading groups of correlated features to share credit (Figure 2.3 conditional). For example, if the feature *weight* is introduced when *BMI* is absent, then conditional Shapley values will only consider values of *BMI* that make sense given the known value of *weight* (i.e., "on-manifold" values); as a consequence, if the model depends on *BMI* but not *weight*, we would still observe that introducing *weight* affects the conditional expectation of the model output. In contrast, although marginal Shapley values perturb the data in less realistic ways ("off-manifold"),

they are able to distinguish between correlated variables and identify whether the model functionally depends on *BMI* or *weight*, which is useful for model debugging [123].

Having two popular types of Shapley value explanations has been cited as a weakness [108], but this issue is not unique to Shapley values; it is encountered by a large number of model explanation methods [42], and it is fundamental to understanding feature importance with correlated or statistically dependent features. For example, with linear models, there are similar issues with handling correlation, where multicollinearity can result in different coefficients with equivalent accuracy. One solution to handle multicollinearity for linear models is to utilize appropriate regularization (e.g., ridge regression) [29], otherwise credit (coefficients) can be split among correlated features somewhat arbitrarily. In the model explanation context, correlated features represent a similar challenge, and the multiple ways of handling absent features can be understood as different approaches to disentangle credit for correlated features.

Another solution to address correlated features is to incorporate causal knowledge. Causal inference approaches typically assume knowledge of an underlying causal graph (i.e., a directed graph where edges indicate that one feature causes another) from which correlations between features arise. There are a number of Shapley value explanation methods that leverage an underlying causal graph, such as causal Shapley values [80], asymmetric Shapley values [67], and Shapley Flow [202]. The major drawback of these approaches is that they assume prior knowledge of causal graphs that are unknown in the vast majority of applications. For this reason, conditional and marginal Shapley values represent more viable options in many practical situations.

In this paper, we advocate for marginal and conditional Shapley values because they are more practical than causal Shapley values, and they avoid the problematic choice of a fixed baseline as in baseline Shapley values. In addition, they cover two of the most common use-cases for Shapley value explanations and model interpretation in general: (1) understanding a model's informational dependencies, and (2) understanding the model's functional form. An important final distinction between marginal and conditional Shapley values is the ease

of estimation. As we discuss next in Section 2.5, marginal Shapley values turn out to be much simpler to estimate than conditional Shapley values.

2.5 Algorithms to estimate Shapley value explanations

Here, we describe algorithmic approaches to address the two main challenges for generating Shapley value explanations: (1) removing features to estimate the coalitional game, and (2) tractably calculating Shapley values despite their exponential complexity.

2.5.1 Feature removal approaches

Previously we introduced three main feature removal approaches and discussed tradeoffs between them. In this section, we discuss how to calculate the coalitional games that correspond to the most popular variants of Shapley value explanations: baseline Shapley values, marginal Shapley values, and conditional Shapley values.

Baseline Shapley values

The coalitional game for baseline Shapley values is defined as

$$v(S) = f(x^e_S, x^b_{\bar{S}}), \tag{2.2}$$

where $f(x^e_S, x^b_{\bar{S}})$ denotes evaluating f on a hybrid sample where present features are taken from the explicand x^e and absent features are taken from the baseline x^b. To compute the value of this coalitional game, we can simply create a hybrid sample and then return the model's prediction for that sample. It is possible to exactly compute this coalitional game, unlike the remaining approaches. The only parameter is the choice of baseline, which can be a somewhat arbitrary decision.

Marginal Shapley values

For marginal Shapley values, the coalitional game is the marginal expectation of the model output,

$$v(S) = \mathbb{E}_{p(x_{\bar{S}})}[f(x^e_S, x_{\bar{S}})], \tag{2.3}$$

where $x_{\bar{S}}$ is treated as a random variable representing the missing features and we take the expectation over the marginal distribution $p(x_{\bar{S}})$ for these missing features.

A natural approach to compute the marginal expectation is to leverage the training or test data to calculate an empirical estimate. A standard assumption in machine learning is that the data are independent draws from the data distribution $p(x)$, so we can designate a set of observed samples E as an empirical distribution and use their values for the absent features (Figure 2.3d Marginal):

$$v(S) = \frac{1}{|E|} \sum_{x^b \in E} f(x^e_S, x^b_{\bar{S}}). \tag{2.4}$$

From Equation (2.4), it is clear that the empirical marginal expectation is the average over the coalitional games for baseline Shapley values with many baselines (Equation (2.2)). As a consequence, marginal Shapley values are also the average over many baseline Shapley values [30]. Due to this, some algorithms estimate marginal Shapley values by first estimating baseline Shapley values for many baselines and then averaging them [123, 30]. Note that marginal Shapley values based on empirical estimates are unbiased if the baselines are drawn i.i.d. from the baseline distribution (e.g., a random subset of rows from the dataset). As such, empirical estimates are considered a reliable way to approximate the true marginal expectation.

The empirical distribution can be the entire training dataset, but in practice it is often a moderate number of samples from the training or test data [188, 123]. The primary parameter is the number of baseline samples and how to choose them. If a large number of baselines is chosen, they can safely be chosen uniformly at random; however, when using a

smaller number of samples, approaches based on k-means clustering can be used to ensure better coverage of the data distribution. This empirical approach also applies to other coalitional games such as the uniform and product of marginals, which are similarly easy to estimate [132].

Conditional Shapley values

For conditional Shapley values, the coalitional game is the conditional expectation of the model output,

$$v(S) = \mathbb{E}_{p(x_{\bar{S}}|x^e_S)}[f(x^e_S, x_{\bar{S}})], \tag{2.5}$$

where $x_{\bar{S}}$ is considered a random variable representing the missing features, and we take the expectation over the conditional distribution $p(x_{\bar{S}} \mid x^e_S)$ of these missing features given the known features x^e_S from the explicand.

Computing conditional Shapley values is more difficult because the required conditional distributions are not readily available from the training data. We can empirically estimate conditional expectations by averaging model predictions from samples that match the explicand's present features (Figure 2.3d conditional), and this exactly estimates the conditional expectations as the number of baseline samples goes to infinity. However, this empirical estimate does not work well in practice: the number of matching rows may be too low in the presence of continuous features or a large number of features, leading to inaccurate and unreliable estimates [188]. For instance, if one conditions on a height of 5.879 feet, there are likely very few individuals with that exact height, so the empirical conditional expectation will average over very few samples' predictions, or potentially just the single prediction from the explicand itself.

One natural solution is to approximate the conditional expectation based on similar feature values rather than exact matches [128, 188, 4]. For instance, rather than condition on baselines that are 5.879 feet tall, we can condition on baselines that are between 5.879 ± 0.025 feet tall. This approach requires a definition of similarity, which is not obvious and may be

an undesirable prerequisite for an explanation method. Furthermore, these approaches do not fully solve the curse of dimensionality, and conditioning on many features can still lead to inaccurate estimates of the conditional expectation.

Instead of empirical estimates, a number of approaches based on fitting models have been proposed to estimate the conditional expectations. Many of these have been identified in the broader context of removal-based explanations [42], but we reiterate them here, summarizing practical strengths and weaknesses:

- **Parametric assumptions.** Chen et al. [29] and Aas et al. [4] assume Gaussian or Gaussian-copula distributions. Conditional expectations for Gaussian random variables have closed-form solutions and are computationally efficient once the joint distribution's parameters have been estimated, but these approaches can have large bias if the parametric assumptions are incorrect.

- **Generative model.** Frye et al. [66] use a conditional generative model to learn the conditional distributions given every subset of features. The generative model provides samples from approximate conditional distributions, and with these we can average model predictions to estimate the conditional expectation. In general, this approach is more flexible than simple parametric assumptions, but it has variance due to the stochastic nature of training deep generative models, and it is difficult to assess whether the generative model accurately approximates the exponential number of conditional distributions.

- **Surrogate model.** Frye et al. [66] use a surrogate model to learn the conditional expectation of the original model given every subset of features. The surrogate model is trained to match the original model's predictions with arbitrarily held-out features, and doing so has been shown to directly approximate the conditional expectation, both for regression and classification models [42]. This approach is as flexible as the generative model, but it has several practical advantages: it is simpler to train, it

requires only one model evaluation to estimate the conditional expectation, and it has been shown to provide a more accurate estimate in practice [66].

- **Missingness during training.** Covert et al. [42] describe an approach for directly estimating the conditional expectation by training the original model to accommodate missing features. Unlike the previous approaches, this approach cannot be applied post-hoc with arbitrary models because it requires modifying the training process.

- **Separate models.** Lipovetsky and Conklin [117], Štrumbelj et al. [185], and Williamson and Feng [208] directly estimate the conditional expectation given a subset of features as the output of a model trained with that feature subset. If every model is optimal (e.g., the Bayes classifier), then the conditional expectation estimate is exact [42]. In practice, however, the various models will be sub-optimal and unrelated to the original one, making it unsatisfying to view it as an explanation for the original model trained on all features. Furthermore, the computational demands of training models with many feature subsets is significant, particularly for non-linear models such as tree ensembles and neural networks.

As we have just shown, there are a wide variety of approaches to model conditional distributions or directly estimate the conditional expectations. These approaches will generally be biased, or inexact, because the coalitional game we require is based on the true underlying conditional expectation. Compounding this, it is difficult to quantify the approximation quality because the conditional expectations are unknown, except in very simple cases (e.g., synthetic multivariate Gaussian data).

Of these approaches, the empirical approach produces poor estimates, parametric approaches require strong assumptions, missingness during training is not model-agnostic, and separate models is not exactly an explanation of the original model. Instead, we believe approaches based on a generative model or a surrogate model are more promising. These approaches are more flexible, but both require fitting an additional deep model. To assess

these deep models, Frye et al. [66] propose two reasonable metrics based on mean squared error of the model's output to evaluate the generative and surrogate model approaches. Future work may include identifying robust architectures/hyperparameter optimization for surrogate and generative models, analyzing how conditional Shapley value estimates change for non-optimal surrogate and generative models, and evaluating bias in conditional Shapley value estimates for data with known conditional distributions.

Some of the approaches we discussed approximate the intermediate conditional distributions (empirical, parametric assumptions, generative model) whereas others directly approximate conditional expectations (surrogate model, missingness during training, separate models). It is worth noting that approaches based on modeling conditional distributions are independent of the particular model f. This suggests that if a researcher fits a high-quality generative model to a popular dataset, then any subsequent researchers can re-use this generative model to estimate conditional Shapley values for their own predictive models. However, even if fit properly, approaches based on modeling conditional distributions may be more computationally expensive, because they require evaluating the model with many generated samples to estimate the conditional expectation. As such, the surrogate model approach may be more effective than the generative model approach in practice [66], and it has been used successfully in recent work [90, 43].

In summary, in order to compute conditional Shapley values, there are *two primary parameters*: (1) the approach to model the conditional expectation, for which there are several choices. Furthermore, within the approaches that rely on deep models (generative model and surrogate model), the training and architecture of the deep model becomes an important yet complex dependency. (2) The baseline set used to estimate the conditional distribution or model the conditional expectation, because each approach requires a set of baselines (e.g., the training dataset) to learn dependencies between features. Different sets of baselines can lead to different scientific questions [132]. For instance, using baselines drawn from older male subpopulations, we can ask "*why does an older male individual have a mortality risk of X% relative to the subpopulation of older males?*" [30].

2.5.2 Tractable estimation strategies

Calculating Shapley values is, in the general case, an NP-hard problem [50, 59]. Intuitively, a brute force calculation based on Equation (2.1) has exponential complexity in the number of features because it involves evaluating the model with all possible feature subsets. Given the long history of Shapley values, there is naturally considerable research into their calculation. Within the game theory literature, two types of estimation strategies have emerged [26, 61, 84]: (1) approximation-based strategies that produce unbiased Shapley value estimates for any game [25], and (2) assumption-based strategies that can produce exact results in polynomial time for specific types of games [131, 74, 25].

These two strategies have also prevailed in Shapley value explanation literature. However, because some approaches rely on both assumptions and approximations, we instead categorize the approaches as *model-agnostic* or *model-specific*. *Model-agnostic* estimation approaches make no assumptions on the model class and often rely on stochastic, sampling-based estimators [183, 121, 146, 90]. In contrast, *model-specific* approaches rely on assumptions about the machine learning model's class to improve the speed of calculation, although sometimes at the expense of exactness [29, 123, 30, 9, 203].

Model-agnostic approaches

There are several types of model-agnostic approaches to estimate Shapley value explanations. In general, these are approximations that sidestep evaluating the model with an exponential number of subsets by instead using a smaller number of subsets chosen at random. These approaches are generally unbiased but stochastic; that is, their results are non-deterministic, but they are correct in expectation.

To introduce these approaches, we describe how each approximation can be tied to a distinct mathematical characterization of the Shapley value. We provided one such characterization in Section 2.3 (Equation (2.1)), but there are multiple equations to represent the Shapley value, and each one suggests its own approximation approach. For simplicity, we

discuss these approaches in the context of a game v where we ignore the choice of feature removal technique (baseline, marginal or conditional).

The classic Shapley value definition is as a **semivalue** [55], where each player's credit is a weighted average of the player's marginal contributions. For this we require a weighting function $P(S)$ that depends only on the subset's cardinality, or where $\sum_{S \subseteq D \setminus \{i\}} P(S) = 1$ for $i = 1, \ldots, d$. Then, the value for player i is given by

$$\phi_i(v) = \sum_{S \subseteq D \setminus \{i\}} P(S)\big(v(S \cup \{i\}) - v(S)\big). \tag{2.6}$$

Castro et al. [25] proposed an unbiased, stochastic estimator (ApproSemivalue) for *any* semivalue (i.e., with arbitrary weighting function) that involves sampling subsets from $D \setminus \{i\}$ with probability given by $P(S)$. In this algorithm, each player's Shapley value is estimated one at a time, or independently. To use ApproSemivalue to estimate Shapley values, we simply have to draw subsets according to the distribution $P(S) = \frac{|S|!(|D|-|S|-1)!}{|D|!}$. While apparently simple, Shapley value estimators inspired directly by the semivalue characterization are uncommon in practice because sampling subsets from $P(S)$ is not straightforward.

Two related approaches are Local Shapley (L-Shapley) and Connected Shapley (C-Shapley) [34]. Unlike other model-agnostic approaches, L-Shapley and C-Shapley are designed for structured data (e.g., images) where nearby features are closely related (spatial correlation). Both approaches are biased Shapley value estimators because they restrict the game to consider only coalitions of players within the neighborhood of the player being explained[4]. They are variance-free for sufficiently small neighborhoods, but for large neighborhoods it may still be necessary to use sampling-based approximations that introduce variance.

Next, the Shapley value can also be viewed as a **random order value** [172, 137], where a player's credit is the average contribution across many possible orderings. Here,

[4]Technically L-Shapley and C-Shapley are *probabilistic values* [137], a generalization of semivalues, because their weighting functions differ based on the spatial location of the current feature.

$\pi : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$ denotes a permutation that maps from each position j to the player $\pi(j)$. Then, $\Pi(D)$ denotes the set of all possible permutations and $Pre^i(\pi)$ denotes the set of predecessors of player i in the order π (i.e., $Pre^i(\pi) = \{\pi(1), \ldots, \pi(j-1)\}$, if $i = \pi(j)$). Then, the Shapley value's random order characterization is the following:

$$\phi_i(v) = \frac{1}{|D|!} \sum_{\pi \subseteq \Pi(D)} (v(Pre^i(\pi) \cup \{i\}) - v(Pre^i(\pi))). \tag{2.7}$$

There are two unbiased, stochastic estimation approaches based on this characterization. The first approach is IME (Interactions-based Method for Explanation) [183], which estimates Equation (2.7) for each player with a fixed number of random permutations from $\Pi(D)$. Perhaps surprisingly, IME is analogous to ApproSemivalue, because identifying the preceding players in a random permutation can be understood as sampling from the probability distribution $P(S)$. One variant of IME improves the estimator's convergence by allocating more samples to estimate $\phi_i(v)$ for players with high variance in their marginal contributions, which we refer to as *adaptive sampling* [184].

The second approach is ApproShapley, which explains all features simultaneously given a set of sampled permutations [25]. Rather than draw permutations independently for each player, this approach iteratively adds all players according to each sampled permutation so that all players' estimates rely on the same number of marginal contributions based on the same permutations. There are many variants that aim to draw *samples efficiently* (i.e., reduce the variance of the estimates): antithetic sampling [164, 135], stratified sampling [126, 26], orthogonal spherical codes [135], and more [196, 135, 84]. Of these approaches, antithetic sampling is the simplest. After sampling a subset and evaluating its marginal contribution, antithetic sampling also evaluates the marginal contribution of the inverse of that subset $(N \setminus S)$. Recent work finds that antithetic sampling provides near-best convergence in practice compared to several more complex methods [135].

The primary difference between IME and ApproShapley is that IME estimates $\phi_i(v)$ independently for each player, whereas ApproShapley estimates them simultaneously for $i =$

$1, \ldots, d$. This means that IME can use adaptive sampling, which evaluates a different number of marginal contributions for each player and can greatly improve convergence when many players have low importance. In contrast, walking through permutations as in ApproShapley is advantageous because (1) it halves the number of evaluations of the game (which are expensive) by reusing them, and (2) it guarantees that the efficiency axiom is satisfied (i.e., the estimated Shapley values sum to the model's prediction).

The third characterization of the Shapley value is as a **least squares value** [28, 166]. In this approach, the Shapley value is viewed as the solution to a weighted least squares (WLS) problem. The problem requires a weighting kernel $W(S)$, and the credits are the coefficients that minimize the following objective,

$$\phi(v) = \arg\min_{\beta} \sum_{S \subseteq D} W(S)(u(S) - v(S))^2, \tag{2.8}$$

where $u(S) = \beta_0 + \sum_{i \in S} \beta_i$ is an additive game[5]. In order to obtain the Shapley value, we require the weighting kernel $W(S) = \frac{|D|-1}{\binom{|D|}{|S|}|S|(|D|-|S|)}$.

Based on this definition, a natural estimation approach is to sample a moderate number of subsets according to $W(S)$ and then solve the approximate WLS problem (Equation (2.8)). This approach is known as *KernelSHAP* [121], and its statistical properties have been studied in recent work: KernelSHAP is consistent and asymptotically unbiased[6] [208], and it is empirically unbiased even for a moderate number of samples [39]. Variants of this approach include (1) a regularized version that introduces bias while reducing variance [121], and (2) an antithetic version that pairs each sampled subset with its complement to improve convergence [39].

Another approach that we refer to as SGD-Shapley is also based on the least squares characterization. It samples subsets according to the weighting kernel $W(S)$, but it iter-

[5]We present a generalized version of the least squares value, which originally involved additional constraints on the coefficients [166].

[6]Williamson and Feng [208] prove this result for a global version of KernelSHAP, but it holds for the original version as well because it is an M-estimator [197].

atively estimates the solution from a random initialization using projected stochastic gradient descent [177]. Although it is possible to prove meaningful theoretical results for this strategy [177], our empirical evaluation shows that the KernelSHAP estimator consistently outperforms this approach in terms of its convergence (i.e., consistently lower estimation error given an equal number of samples).

Finally, the last approach based on the least squares characterization is FastSHAP [90, 43]. FastSHAP learns a separate model (an *explainer*) to estimate Shapley values in a single forward pass, and it is trained by amortizing the WLS problem (Equation (2.8)) across many data examples. As a consequence of its WLS training objective, the globally optimal estimation model is a function that outputs exact Shapley values. Since the explanation model will generally be non-optimal, the resultant estimates offer imperfect accuracy and are random across separate training runs [90]. The major advantage of FastSHAP is that developers can frontload the cost of training the explanation model, thereby providing subsequent users with fast Shapley value explanations.

The fourth characterization of the Shapley value is based on a **multilinear extension** of the game [148, 146]. The multilinear extension extends a coalitional game to be a function on the d-cube $[0,1]^d$ that is linear separately in each variable. Based on an integral of the multilinear extension's partial derivatives, the Shapley value can be defined as

$$\phi_i(v) = \int_0^1 g_i(q)dq, \tag{2.9}$$

where $g_i(q) = \mathbb{E}[v(G_i \cup \{i\}) - v(G_i)]$ and G_i is a random subset of $D \setminus \{i\}$, with each feature having probability q of being included. Perhaps surprisingly, as with the random order value characterization, estimating this formulation involves averaging many marginal contributions where the subsets are effectively drawn from $P(S) = \frac{|S|!(|D|-|S|-1)!}{|D|!}$.

Based on this characterization, Okhrati and Lipani [146] introduced an unbiased sampling-based estimator that we refer to as multilinear extension sampling. The estimation consists of (1) sampling a q from the range $[0,1]$, and then (2) sampling random subsets based on q

and evaluating the marginal contributions. This procedure introduces an additional parameter, which is the balance between the number of samples of q and the number of subsets E_i to generate for each value of q. The original version draws 2 random subsets for each q, where q is sampled at fixed intervals according to the trapezoid rule [146]. Finally, in terms of variants, Okhrati and Lipani [146] find that antithetic sampling improves convergence, where for each subset they also compute the marginal contribution for the inverse subset.

To summarize, there are three main characterizations of the Shapley value from which unbiased, stochastic estimators have been derived: random order values, least squares values, and multilinear extensions. Within each approach, there are a number of variants. (1) *Adaptive sampling*, which has only been applied to IME (per-feature random order), but can easily be applied to a version of multilinear extension sampling that explains features independently. (2) *Efficient sampling*, which aims to carefully draw samples to improve convergence over independent sampling. In particular, one version of efficient sampling, antithetic sampling, is easy to implement and effective; it has been applied to ApproShapley, KernelSHAP, and multilinear extension sampling, and it can also be easily extended to IME[7]. Although the other efficient sampling techniques have mainly been examined in the context of ApproShapley, similar benefits may exist for IME, KernelSHAP, and multilinear extension sampling. Finally, there is (3) *amortized explanation models*, which have only been applied to the least squares characterization [90], but may be extended to other characterizations that where the Shapley value can be viewed as the solution to an optimization problem.

Empirically comparing model-agnostic approaches

In Figure 2.5, we examine the convergence of the four main stochastic estimators (IME, ApproShapley, KernelSHAP and multilinear extension sampling) as well as several popular

[7]The approach of drawing an inverse subset for each independently drawn subset was introduced as "paired sampling" in KernelSHAP [39], "antithetic sampling" for ApproShapley [135], and "halved sampling" for multilinear extension sampling [146]. We refer to all of these approaches as "antithetic sampling." [164]

variants of these approaches (adaptive and antithetic sampling[8]). We experiment with three datasets with varying numbers of features, all using XGBoost models. To calculate convergence, we calculate estimation error (mean squared error) of estimates of baselines Shapley values from the true baseline Shapley values computed using Interventional TreeSHAP. We also introduce four new approaches: IME with antithetic sampling, a new version of multilinear extension sampling that explain one feature at a time[9], and antithetic/adaptive sampling variants of this new multilinear extension sampling approach.

For the diabetes and NHANES datasets, which have 10 and 79 features respectively, the anthithetic version of KernelSHAP converges fastest to the true Shapley values. However, for the blog dataset, which has 280 features, we observe that IME and multilinear (feature) converge fastest, likely due to their use of adaptive sampling. Based on this finding, we hypothesize that adaptive sampling is important in the presence of many features, because with many features there are more likely to be features without interaction effects or with near-zero contributions. Such features will have little to no variance in their marginal contributions. We tested this hypothesis by modifying the diabetes dataset by adding 100 zero features that have no impact on the model and therefore no interaction effects. In this scenario, we find that the dummy features slow convergence and that the adaptive sampling approaches are least affected, likely because they can determine which features converge rapidly and then ignore them.

Finally, an important desideratum of feature attribution estimates is the *efficiency axiom* [172], which means that the attributions sum to the game's value with all players minus the value with none (Figure 2.2b). In terms of the sampling-based estimators, only ApproShapley and KernelSHAP are guaranteed to satisfy the efficiency property. However, it is possible to adjust attributions by evenly splitting the efficiency gap between all features using the *additive efficient normalization* operation [166]. This normalization step is used to

[8]The other efficient sampling techniques and amortized explanation models are more complex and out of the scope for this review.

[9]The initial version of multilinear extension sampling [146] explains all features simultaneously. This enables re-use of model evaluations, but prohibits the use of adaptive sampling.

ensure that the efficiency property holds for both FastSHAP and SGD-Shapley [90, 177]. It can also be used to ensure efficiency as a final post-processing step for IME and multilinear extension sampling, and it is guaranteed to improve estimates in terms of Euclidean distance to the true Shapley values without affecting the estimators' bias [90].

These model-agnostic strategies for estimating Shapley values are appealing because they are flexible: they can be applied to any coalitional game and therefore any machine learning model. However, one major downside of these approaches is that they are inherently stochastic. Although most methods are guaranteed to be correct given an infinite number of samples (i.e., they are consistent estimators), users have finite computational budgets, leading to estimators with potentially non-trivial variance. In response, some methods utilize techniques to forecast and detect convergence when the estimated variance drops below a fixed threshold [41, 39]. However, even with convergence detection, model-agnostic explanations can be prohibitively expensive. Motivated in part by the computational complexity of these methods, a number of approaches have been developed to estimate Shapley value explanations more efficiently by making assumptions about the type of model being explained.

Model-specific approaches

In terms of model-specific approaches, algorithms have been designed for several popular model types: linear models, tree models, and deep models. These approaches are less flexible than model-agnostic approaches, in that they assume a specific model type and often a specific feature removal approach, but they are generally significantly faster to calculate.

First, one of the simplest model types to explain is **linear models** (also discussed in Section 2.4). For linear models, baseline and marginal Shapley values have a closed-form solution based on the coefficients of the linear model (β) and the values of the explicand (x^e) and the baseline(s). If we let E represent a set of baseline values for marginal Shapley values, or $E = \{x^b\}$ for baseline Shapley values, we can write the mean value for feature x_i as $\mu_i = \frac{1}{|E|} \sum_{x^b \in E} x^b$. Then, LinearSHAP [121, 184] gives the following result for the marginal or baseline Shapley values:

$$\phi_i(f, x^e) = \beta_i(x_i^e - \mu_i^b). \tag{2.10}$$

Computing these values is straightforward, and it has linear complexity in the number of features, versus exponential complexity in the general case. Interestingly, for linear models the marginal Shapley value is equivalent to the baseline Shapley value with a mean baseline, which is why the mean baseline may be a good choice for linear models (see Section 2.4.4).

Alternatively, correlated LinearSHAP [29] estimates conditional Shapley values for linear models assuming that the data follows a *multivariate Gaussian distribution*. The key idea behind this approach is that for linear models, the expectation of the model's prediction equals the model's prediction for the average sample, or $\mathbb{E}[f(x) \mid x_S] = f(\mathbb{E}[x \mid x_S]))$, which is not true in general. Using the closed-form solution for $\mathbb{E}[x \mid x_S]$ given multivariate normal data, we can calculate the conditional Shapley values as

$$\phi_i(f, x^e) = \beta^\top A_i \mu + \beta^\top B_i x^e, \tag{2.11}$$

where μ is the mean feature vector and A_i and B_i are summations over an exponential number of coalitions (see [29] for more details). In practice, A_i and B_i are estimated using a sampling-based procedure. Then, subsequent explanations are extremely fast to compute regardless of the explicand value. This approach resembles FastSHAP [90] because both approaches incur most of their computation up-front and then provide very fast explanations. Correlated LinearSHAP is biased if the data does not follow a multivariate Gaussian distribution, which is rarely the case in practice, and it is stochastic because A_i and B_i are estimated.

Next, another popular model type is **tree models**. Tree models include decision trees, as well as ensembles like random forests and gradient boosted trees. These are more complex than linear models and they can represent non-linear and interaction effects, but perhaps surprisingly, it is possible to calculate baseline and marginal Shapley values exactly (Interventional TreeSHAP) and approximate conditional Shapley values (Path-dependent TreeSHAP) tractably for such models.

Interventional TreeSHAP is an algorithm that exactly calculates baseline and marginal Shapley values in time linear in the size of the tree model and the number of baselines [123]. This is possible because a tree model can be represented as a disjoint set of outputs for each leaf in the tree. Then, each leaf's contribution to any given feature's Shapley value can be computed at the leaves of the tree assuming a coalitional game whose players are the features along the path from the root to the current leaf. Using a dynamic programming algorithm, Interventional TreeSHAP computes the Shapley value explanations for all features simultaneously by iterating through the nodes in the tree.

Then, Path-dependent TreeSHAP is an algorithm designed to estimate conditional Shapley values, where the conditional expectation is approximated by the structure of the tree model [123]. Given a set of present features, the algorithm handles internal nodes for absent features by traversing each branch in proportion to how many examples in the dataset follow each direction. This algorithm can be viewed as an application of Shapley cohort refinement [128], where the cohort is defined by the preceding nodes in the tree model and the baselines are the entire training set. In the end, it is possible to estimate a biased, variance-free version of conditional Shapley values in $O(LH^2)$ time, where L is the number of leaves and H is the depth of the tree. Path-dependent TreeSHAP is a biased estimator for conditional Shapley values because its estimate of the conditional expectation is imperfect.

In comparison to Interventional TreeSHAP, Path-dependent TreeSHAP does not have a linear dependency on the number of baselines, because it utilizes node weights that represent the portion of baselines that fall on each node (based on the splits in the tree). Finally, in order to incorporate a tree ensemble, both approaches calculate explanations separately for each tree in the ensemble and then combine them linearly. This yields exact estimates for baseline and marginal Shapley values, because the Shapley value is additive with respect to the model [123].

Finally, another popular but opaque class of models are **deep models** (i.e., deep neural networks). Unlike for linear and tree models, we are unaware of any approach to estimate conditional Shapley values for deep models, but we discuss several approaches that estimate

baseline and marginal Shapley values.

One early method to explain deep models, called DeepLIFT, was designed to propagate attributions through a deep network for a single explicand and baseline [175]. DeepLIFT propagates activation differences through each layer in the deep network, while maintaining the Shapley value's efficiency property at each layer using a chain rule based on either a Rescale rule or a RevealCancel rule, which can be viewed as approximations of the Shapley value [30]. Due to the chain rule and these local approximations, DeepLIFT produces biased estimates of baseline Shapley values. Later, an extension of this method named DeepSHAP was designed to produce biased estimates of marginal Shapley values [30]. Despite its bias, DeepSHAP is useful because the computational complexity is on the order of the size of the model and the number of baselines, and the explanations have been shown to be useful empirically [158, 30]. In addition, the Rescale rule is general enough to propagate attributions through pipelines of linear, tree, and deep models [30].

Another method to estimate baseline Shapley values for deep models is Deep Approximate Shapley Propagation (DASP) [9]. DASP utilizes uncertainty propagation to estimate baseline Shapley values. To do so, the authors rely on a definition of the Shapley value that averages the expected marginal contribution for each coalition size. For each coalition size k and a zero baseline, the input distribution from the random coalitions is modeled as a normal random variable whose parameters are a function of k. Since the input distributions are normal random variables, it is possible to propagate uncertainty for specific layers by matching first and second-order central moments and thereby estimate each expected marginal contribution. Based on an empirical study, DASP produces baseline Shapley values estimates with lower bias than DeepLIFT [9]. However, DASP is more computationally costly and requires up to $O(d^2)$ model evaluations, where d is the number of features. Although DASP is deterministic (variance-free) with $O(d^2)$ model evaluations, it is biased because the moment propagation relies on an assumption of independent inputs that is violated at internal nodes whose inputs are given by the previous layer's outputs.

One final method to estimate baseline Shapley values for deep models is Shapley Ex-

planation Networks (ShapNets) [203]. ShapNets restrict the deep model to have a specific architecture for which baseline Shapley values are easier to estimate. The authors make a stronger assumption than DASP or DeepLIFT/DeepSHAP by not only restricting the model to be a neural network, but by requiring a specific architecture where hidden nodes have a small input dimension h (typically between 2-4). In this setting, ShapNets can construct baseline Shapley values for each hidden node because the exponential cost is low for small h. The authors present two methods that follow the architecture assumption. (1) *Shallow ShapNets*: networks that have a single hidden layer, and where baseline Shapley values can be calculated exactly. Although they are easy to explain, these networks suffer in terms of model capacity and have lower predictive accuracy than other deep models. (2) *Deep ShapNets*: networks with multiple layers through which we can calculate explanations hierarchically. For Deep ShapNets, the final estimates are biased because of this hierarchical, layer-wise procedure. However, since Deep ShapNets can have multiple layers, they are more performant in terms of making predictions, although they are still more limited than standard deep models. An additional advantage of ShapNets is that they enable developers to regularize explanations based on prior information without a costly estimation procedure [203].

DASP and Deep ShapNets are originally designed to estimate baseline Shapley values with a zero baseline: DASP assumes a zero baseline to obtain an appropriate input distribution, and Deep ShapNets uses zero baselines in internal nodes. However, it may be possible to adapt DASP and Deep ShapNets to use arbitrary baselines (as in DeepLIFT and Shallow ShapNets), in which case it would be possible to estimate marginal Shapley values as DeepSHAP does. In terms of computational complexity, DeepLIFT, Shallow ShapNets, and Deep ShapNets can estimate baseline Shapley values with a constant number of model evaluations (for a fixed h). In contrast, DASP requires a minimum of d model evaluations and up to $O(d^2)$ model evaluations for a single estimate of baseline Shapley values.

A final difference between these approaches is in their assumptions. Shallow ShapNets and Deep ShapNets make the strongest assumptions by restricting the deep model's architecture. DASP makes a strong assumption that we can perform first and second-order central moment

matching for each layer in the deep model, and the original work only describes moment matching for affine transformations, ReLU activations and max pooling layers. Finally, DeepLIFT and DeepSHAP assume deep models, but they are flexible and support more types of layers than DASP or ShapNets. However, as a consequence of DeepLIFT's flexibility, its baseline Shapley value estimates have higher bias compared to DASP or ShapNets [9, 203].

2.6 Discussion

In this work, we provided a detailed overview of numerous algorithms for generating Shapley value explanations. In particular, we delved into the two main factors of complexity underlying such explanations: the feature removal approach and the tractable estimation strategy. Disentangling the complexity in the literature into these two factors allows us to more easily understand the key innovations in recently proposed approaches.

In terms of feature removal approaches, algorithms that aim to estimate baseline Shapley values are generally unbiased, but choosing a single baseline to represent feature removal is challenging. Similarly, algorithms that aim to estimate marginal Shapley values will also generally be unbiased in their Shapley value estimates. Finally, algorithms that aim to estimate conditional Shapley values will be biased because the conditional expectation is fundamentally challenging to estimate. Conditional Shapley values are currently difficult to estimate with low bias and variance, except in the case of linear models; however, depending on the use case, it may be preferable to use an imperfect approximation rather than switch to baseline or marginal Shapley values.

In terms of the exponential complexity of Shapley values, model-agnostic approaches are often more flexible and bias-free, but they produce estimators with non-trivial variance. By contrast, model-specific approaches are typically deterministic and sometimes unbiased. Of the model-specific methods, only LinearSHAP and Interventional TreeSHAP have no bias for baseline and marginal Shapley values. In particular, we find that the Interventional TreeSHAP explanations are fairly remarkable for being non-trivial, bias-free, and variance-free. As such, tree models including decision trees, random forests, and gradient boosted

trees are particularly well-suited to Shapley value explanations.

Furthermore, based on the feature removal approach and estimation strategy of each approach, we can understand the sources of bias and variance within many existing algorithms (Table 2.1). IME [183], for instance, is bias-free, because marginal Shapley values and the random order value estimation strategy are both bias-free. However, IME estimates have non-zero variance because the estimation strategy is stochastic (random order value sampling). In contrast, Shapley cohort refinement estimates [128] have both non-zero bias and non-zero variance. Their bias comes from modeling the conditional expectation using an empirical, similarity-based approach, and the variance comes from the sampling-based estimation strategy (random order value sampling).

In practice, Shapley value explanations are widely used in both industry and academia. Although they are powerful tools for explaining models, it is important for users to be aware of important parameters associated with the algorithms used to estimate them. In particular, we recommend that any analysis based on Shapley values should report parameters including the type of Shapley value explanation (the feature removal approach), the baseline distribution used to estimate the coalitional game, and the estimation strategy. For sampling-based strategies, it is important for users to include a discussion of convergence in order to validate their feature attribution estimates. Finally, developers of Shapley value explanation tools should strive to be transparent about convergence by explicitly performing automatic convergence detection. Convergence results based on the central limit theorem are straightforward for the majority of model-agnostic estimators we discussed, although they are not always implemented in public packages. Note that convergence analysis is more difficult for the least squares estimators, but Covert and Lee [39] discuss this issue and present a convergence detection approach for KernelSHAP.

Future research directions include investigating new stopping conditions for convergence detection. Existing work proposes stopping once the largest standard deviation is smaller than a prescribed threshold [39], but depending on the threshold, the variance may still be high enough that the relative importance of features can change. Therefore, a new stopping

condition could be when additional marginal contributions are highly unlikely to change the relative ordering of attributions for all features. Another important future research direction is Shapley value estimation for deep models. Current model-specific approaches to explain deep models are biased, even for marginal Shapley values, and no model-specific algorithms exist to estimate conditional Shapley values. One promising model-agnostic approach is FastSHAP [90, 43], which speeds up explanations using an explainer model, although it requires a large upfront cost to train this model. Finally, because approximating the conditional expectation for conditional Shapley values is so hard, it constitutes an important future research direction that would benefit from new methods or systematic evaluations of existing approaches.

2.7 Recommendations based on data domain

What we have discussed in this paper is largely agnostic to the type of data. However, there are a few characteristics of the data being analyzed that may change the best practices for generating Shapley value explanations.

The first characteristic is the number of features. Models with more input features will be more computationally expensive to explain. In particular, for model-agnostic algorithms, as the number of features increases, the total number of possible coalitions increases exponentially. In this setting it may be valuable to reduce the number of features by carefully filtering the ones which do not vary or are highly redundant with other features. This type of feature selection can even be performed prior to model-fitting and is already a common practice.

The second characteristic is the number of samples. The number of samples likely plays a larger role in fitting the original predictive model than in generating the explanation. However, for conditional Shapley values, having a large number of samples is important for creating accurate estimates of the conditional expectations/distributions. If the number of samples is very low, it may be better to rely on parametric assumptions or use marginal Shapley values instead.

The third characteristic is the feature correlation. In highly correlated settings, one can expect larger discrepancies between marginal and conditional Shapley values. Then, carefully choosing the feature removal approach or comparing estimates of both marginal and conditional Shapley values can be valuable. Highly correlated features can also make it harder to understand feature attributions in general. For images in particular, the importance of a single pixel may not be semantically meaningful in isolation. In these cases, it may be useful to use explanation methods that aim to understand higher level concepts [103].

Beyond correlated features, there may also be structure within the data. Tabular data are typically considered unstructured whereas image and text data is structured because neighboring pixels and words are strongly correlated. For tabular data, it may be best to use tree ensembles (gradient boosted trees or random forests) which are both performant and easy to explain using the two versions fo TreeSHAP (Interventional and Path-dependent) [123]. For structured data, it may be valuable to use methods such as L-Shapley and C-Shapley that are designed to estimate Shapley value explanations more tractably in structured settings [34]. Furthermore, grouping features may be natural for structured data (e.g., superpixels for image data or sentences/n-grams for text data), because it greatly reduces the computational complexity of most algorithms. Finally, since structured data often calls for complex deep models, which are expensive to evaluate, methodologies such as FastSHAP can be useful for accelerating explanations.

The fourth characteristic is prior knowledge of causal relationships. Although causal knowledge is unavailable for the vast majority of datasets, it can be used to generate Shapley value explanations that better respect causal relationships [80, 67] or generate explanations that assign importance to edges in the causal graph [202]. These techniques may be a better alternative to conditional Shapley values, which respect the data manifold, because they respect the causal relationships underlying the correlated features.

The fifth characteristic is whether there is a natural interpretation of absent features. For certain types of data, there may be preconceived notions of feature absence. For instance, in text data it may be natural to remove features from models that take variable length

inputs or use masking tokens to denote feature removal. In images, it is often common to assume some form of gray, black, or blurred baseline; these approaches are somewhat dissatisfying because they are data-specific notions of feature removal. However, given that model evaluations are often exorbitantly expensive in these domains, these techniques may provide simple, yet tractable alternatives to marginal or conditional Shapley values[10].

2.8 Related work

In this paper, we focused on describing popular algorithms to estimate local feature attributions based on the Shapley value. However, there are a number of adjacent explanation approaches that are not the focus of this discussion. Two broad categories of such approaches include alternative definitions of coalitional games, and different game-theoretic solution concepts.

We focus on three popular coalitional games where the players represent features and the value is the model's prediction for a single example. However, as discussed by Covert et al. [42], there are several methods that use different coalitional games, including global feature attributions where the value is the model's mean test loss [41], and local feature attributions where the value is the model's per-sample loss [123]. Other examples include games where the value is the maximum flow of attention weights in transformer models [57], where the players are analogous to samples in the training data [70], where the players are analogous to neurons in a deep model [71], and where the players are analogous to edges in a causal graph [202]. Although these methods are largely outside the scope of this paper, a variety of applications of the Shapley value in machine learning are discussed in Rozemberczki et al. [163].

Secondly, there are game-theoretic solution concepts beyond the Shapley value that can be utilized to explain machine learning models. The first method, named asymmetric Shapley

[10]Note that for the conditional expectation, surrogate models are tractable once they are trained [66, 90] because they directly estimate the conditional expectation in a single model evaluation. However, using a surrogate requires training or fine-tuning an additional model.

values, is designed to generate feature attributions that incorporate causal information [67]. To do so, asymmetric Shapley values are based on random order values where weights are set to zero if they are inconsistent with the underlying causal graph. Next, L-Shapley and C-Shapley (also discussed in Section 2.5.2) are computationally efficient estimators for Shapley value explanations designed for structured data; they are technically *probabilistic values*, a generalization of semivalues [34, 137]. Similarly, Banzhaf values, an alternative to Shapley values, are also semivalues, but each coalition is given equal weight in the summation (see Equation (2.1)). Banzhaf values have been used to explain machine learning models in a variety of settings [96, 33]. Another solution concept designed to incorporate structural information about coalitions is the Owen value. The Owen value has been used to design a hierarchical explanation technique named PartitionExplainer within the SHAP package[11] and as a way to accommodate groups of strongly correlated features [134]. Finally, Aumann-Shapley values are an extension of Shapley values to infinite games [11], and they are connected to an explanation method named Integrated Gradients [189]. Integrated Gradients requires gradients of model's prediction with respect to the features, so it cannot be used for certain types of non-differentiable models (e.g., tree models, nearest neighbor models), and it represents features' absence in a continuous rather than discrete manner that typically requires a fixed baseline (similar to baseline Shapley values).

[11] https://shap.readthedocs.io/en/latest/generated/shap.explainers.Partition.html

Method	Factors of complexity			Properties		
	Estimation strategy	Removal approach	Removal variant	Model-agnostic	Bias-free	Variance-free
ApproSemivalue [25]	SV	None	Exact	Yes	Yes	No
L-Shapley [34]	SV	Marginal	Empirical	Yes	No	No♣
C-Shapley [34]	SV	Marginal	Empirical	Yes	No	No♣
ApproShapley [25]	RO	None	Exact	Yes	Yes	No
IME [183]	RO	Marginal	Empirical	Yes	Yes	No
CES [188]	RO	Conditional	Empirical	Yes	No	No
Shapley cohort refinement [128]	RO	Conditional	Empirical*	Yes	No	No
Generative model [66]	RO	Conditional	Generative	Yes	No	No
Surrogate model [66]	RO	Conditional	Surrogate	Yes	No	No
Multilinear extension sampling [146]	ME	Marginal	Empirical	Yes	Yes♢	No
SGD-Shapley [177]	WLS	Baseline	Exact	Yes	No♡	No
KernelSHAP [121, 39]	WLS	Marginal	Empirical	Yes	Yes♠	No
Parametric KernelSHAP [4]	WLS	Conditional	Parametric	Yes	No	No
Nonparameteric KernelSHAP [4]	WLS	Conditional	Empirical*	Yes	No	No
FastSHAP [90]	WLS	Conditional	Surrogate	Yes	No	No
LinearSHAP [29]	Linear	Marginal	Empirical	No	Yes	Yes
Correlated LinearSHAP [29]	Linear	Conditional	Parametric	No	No	No
Interventional TreeSHAP [123]	Tree	Marginal	Empirical	No	Yes	Yes
Path-dependent TreeSHAP [123]	Tree	Conditional	Empirical*	No	No	Yes
DeepLIFT [175]	Deep	Baseline	Exact	No	No	Yes
DeepSHAP [121]	Deep	Marginal	Empirical	No	No	Yes
DASP [9]	Deep	Baseline	Exact	No	No	No♣
Shallow ShapNet [203]	Deep	Baseline	Exact	No	Yes	Yes
Deep ShapNet [203]	Deep	Baseline	Exact	No	No	Yes

Table 2.1: Methods to estimate Shapley value explanations. We order approaches based on whether or not they are model-agnostic. Then, there are two factors of complexity. The first is the estimation strategy to handle the exponential complexity of Shapley values. For the model-agnostic approaches, the strategies include semivalue (SV), random order value (RO), multilinear extension (ME), and least squares value (LS). Note that the model-agnostic estimation strategies can generally be adapted to apply for any removal approach. For model-specific approaches, the strategies differ for linear, tree, and deep models. Then, the second factor of complexity is the feature removal approach which determines the type of Shapley value explanation (Section 2.5.1). "Any" denotes that it was introduced in game theory, and not for the sake of explaining a machine learning model. Then, we describe the specific removal variant employed by each algorithm. Baseline Shapley values are always computed exactly (Section 2.5.1), marginal Shapley values are always estimated empirically (Section 2.5.1), and conditional Shapley values have a variety of estimation procedures (Section 2.5.1). *These empirical estimates also involve defining a similarity metric. Finally, we report whether approaches are bias-free and/or variance-free. ♢Multilinear extension sampling is unbiased when sampling q uniformly. However, it is more common to use the trapezoid rule to determine q which improves convergence, but can lead to higher bias empirically at smaller numbers of subsets. ♡SGD-Shapley is consistent, but based on our empirical analysis it has high bias relative to other approaches. ♠One version of KernelSHAP has been proven to be bias-free and the original version is asymptotically unbiased [208], although empirically it also appears to be unbiased for moderate numbers of samples [39]. ♣These approaches can be deterministic with a polynomial number of model evaluations, but are often run with fewer evaluations for computational speed.

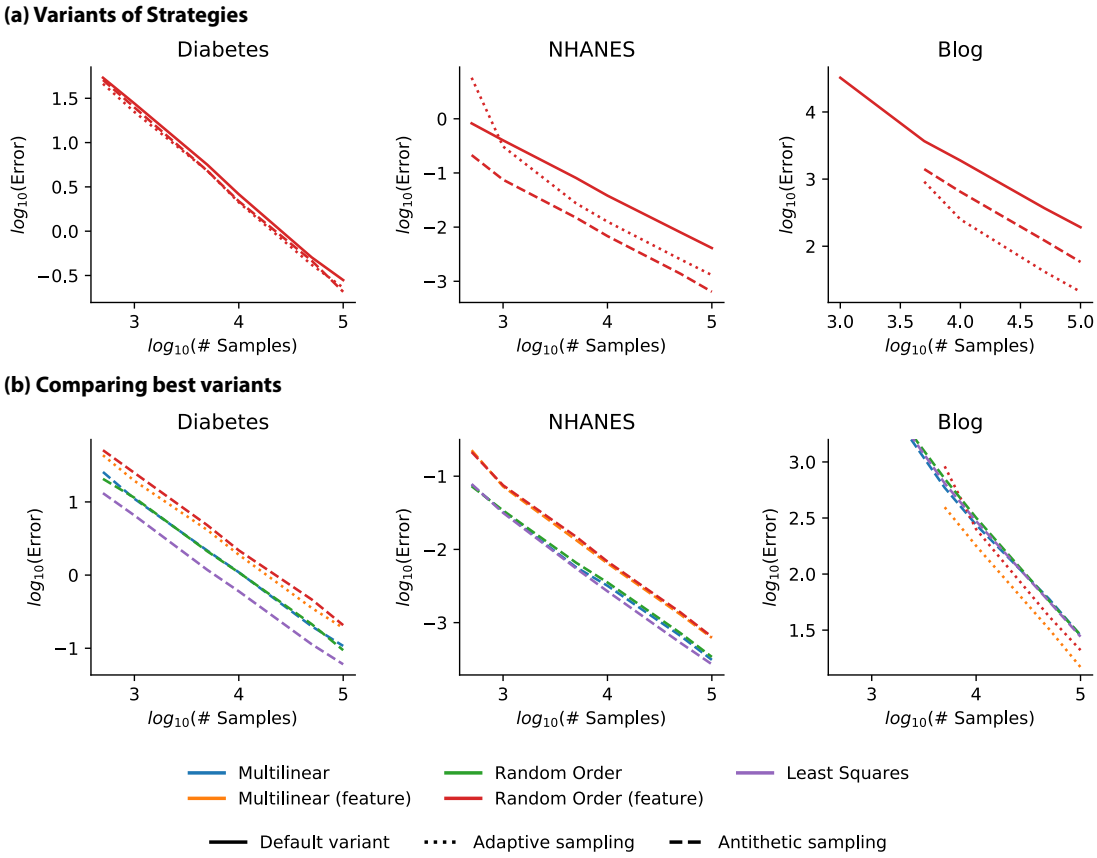


Figure 2.5: Benchmarking unbiased, model-agnostic algorithms to estimate baseline Shapley values for a single explicand and baseline on XGB models with 100 trees. For simplicity, we calculate baseline Shapley values for all methods because we aim to evaluate the tractable estimation strategy rather than the feature removal approach. In particular, the stochastic estimators include each sampling-based approach, where multilinear, random order, random order (feature), and least squares correspond to multilinear extension sampling [146], ApproShapley [25], IME [183], and KernelSHAP [121] respectively. Multilinear (feature) is a new approach based on the multilinear extension sampling approach which explains one feature at a time. In addition, some methods are variants: either antithetic or adaptive sampling. On the x-axis we report the number of samples (subsets) used for each estimate, and on the y-axis we report the MSE relative to the true baseline Shapley value for 100 estimates with that many samples. We use three real-world datasets: diabetes (10 features, regression), NHANES (79 features, classification), blog (280 features, regression). For some variants, no error is shown for small numbers of samples; this is because each approach requires a different minimum number of samples to produce estimates for each feature. (a) Variants of the random order, feature-wise strategy. (b) Benchmarking the most competitive variant of each stochastic estimator chosen according to the lowest error for 10^5 samples. Note that the full blog error plot is truncated to better showcase differences.

Chapter 3

EXPLAINING LINEAR MODELS

3.1 Introduction

One of the most popular approaches to machine learning interpretability in recent years has involved using Shapley values to attribute importance to features [184, 121]. The Shapley value is a concept from coalitional game theory that fairly allocates the surplus generated by the grand coalition in a game to each of its players [172]. In this general sense, the Shapley value allocated to a player i is defined as:

$$\phi_i = \frac{1}{|N|!} \sum_R [v(S^R \cup i) - v(S^R)], \tag{3.1}$$

where R is one possible permutation of the order in which the players join the coalition, S^R is the set of players joining the coalition *before* player i, and $v : S \in \mathcal{P}(N) \rightarrow \mathbb{R}^1$ is a coalitional game that maps from the power set $\mathcal{P}$ of all players N to a scalar value.

While the Shapley value is provably the unique solution which satisfies a variety of axioms for an abstract n-person game, figuring out how to represent a machine learning model (f) as a coalitional game (v) is non-trivial. Previous work has suggested a variety of different functional forms for v, for tasks like data valuation and global feature importance [70, 41]. In this work, we focus on local feature attribution – trying to understand how much each feature contributed to the output of a model for a particular sample. For this application, the reward of the game is typically the conditional expectation of the model's output, where the players in the game are the known features in the conditional expectation. There are two ways the model's output ($f : x \in \mathbb{R}^{|N| \times 1} \rightarrow \mathbb{R}^1$) for a particular sample is used to define

$v(S)$:

1. Conditional expectation: This is the formulation in [121, 4, 67]. The coalitional game is

$$v(S) = \mathbb{E}[f(x)|S] \tag{3.2}$$

where conditioning on S means considering the input X to be a random variable where the features in S are known ($\mathbb{E}[f(X)|X_S = x_S]$).

2. Interventional conditional (marginal) expectation: This approach is endorsed by [87, 188, 47], and in practice is used to approximate the conditional expectation in [121]. Here the coalitional game is defined as:

$$v(S) = \mathbb{E}[f(x)|do(S)] \tag{3.3}$$

where we "intervene" on the features by breaking the dependence between features in S and the remaining features. We refer to Shapley values obtained with either approach as either conditional or marginal Shapley values.

Previous work has pointed out issues with each choice of value function. For example, Janzing et al. [87] and Sundararajan and Najmi [188] both point out that the conditional Shapley value can attribute importance to *irrelevant features* – features which were not used by the model. While this does not violate the original Shapley axioms, it does violate a new axiom called *Dummy* proposed by Sundararajan and Najmi [188], which requires that a feature i will get attribution $\phi_i = 0$, if for any two values x_i and x_i' and for every value $x_{N\setminus i}$, $f(x_i; x_{N\setminus i}) = f(x_i'; x_{N\setminus i})$. On the other hand, papers like Frye et al. [67] have noted that using the marginal Shapley value (which breaks the dependence between features) will lead to evaluating the model on "impossible data points" that lie off the true data manifold.

While recent work has gone so far as to suggest that having two separate approaches presents an irreconcilable problem with using Shapley values for feature attribution [108], in this paper, we argue that rather than representing some critical flaw in using the Shapley value for feature attribution, each approach is meaningful when applied in the proper context.

Further, we argue that this choice depends on whether you want attributions that reflect the behavior of a particular model (true to the model), or attributions that reflect the correlations in the data (true to the data).

3.2 Linear SHAP

In order to understand both approaches, we will focus on *linear models* where we present a novel algorithm to compute the conditional Shapley values. Moving forward, $f(x) = \beta x + b$ where $\beta \in \mathbb{R}^{1\times|N|}$ is a row vector and $b \in \mathbb{R}^1$ a scalar.

3.2.1 Marginal expectation

For a marginal expectation, the Shapley values (which we denote as $\phi_i(f, x)$) are:

$$\phi_i(f, x) = \beta_i(x_i - \mu_i) \tag{3.4}$$

This was shown for independent features [4] and the marginal expectation gives the same explanations.

3.2.2 Conditional expectation

Computing the Shapley values for an conditional expectation is substantially harder, with a number of proposed algorithms for doing so. Sundararajan and Najmi [188] utilizes the empirical distribution, which often assigns zero probability to plausible samples even for large samples. Mase et al. [128] extends this empirical distribution by including a similarity metric. In Aas et al. [4], the unknown features are sampled from either a multivariate gaussian conditional, a gaussian copula conditional, or an empirical conditional distribution. In Frye et al. [67], the conditional distribution is modeled using an autoencoder. For a linear model, the problem reduces to estimating the conditional expectation of x given different

subsets[1]:

$$\phi_i(f,x) = \frac{1}{|N|!}\sum_R \mathbb{E}[f(x) \mid x_{S^R \cup i}] - \mathbb{E}[f(x) \mid x_{S^R}], \tag{3.5}$$

$$= \beta\frac{1}{|N|!}\sum_R \mathbb{E}[x \mid x_{S^R \cup i}] - \mathbb{E}[x \mid x_{S^R}]. \tag{3.6}$$

Estimating this conditional expectation is hard in general, so we assume the inputs $x \sim \mathcal{N}(\mu, \Sigma)$ are multivariate normal. Then, denote the projection matrix that selects a set S as $P_S \in (0,1)^{|S|\times|N|}$ (therefore, $P_S x \in \mathbb{R}^{|S|\times 1}$ returns the features from x in S), then $\mathbb{E}[x \mid x_S]$ is[2]:

$$\overbrace{[P_{\bar{S}}\mu + P_{\bar{S}}\Sigma P_S^T(P_S\Sigma P_S^T)^{-1}(P_S x - P_S\mu)]}^{\text{Conditional expectation for } x_i \in S}\underbrace{P_{\bar{S}}}_{\text{Project to } \mathbb{R}^{|N|}} + \overbrace{xP_S^T P_S}^{\text{Zero } x_i \notin S}. \tag{3.7}$$

At this point, we have a natural solution to obtain the conditional expectation Shapley value for a single sample. If we compute (Equation (3.7)) for all sets S^R, we can use the combinations definition of Shapley values ($\sum_{S\in N\setminus i} W(S,N)(\mathbb{E}[x|x_{S\cup i}] - \mathbb{E}[x|x_S])$) to compute the Shapley value exactly.

Computational complexity: Each term in the summation requires a matrix multiplication/inversion which is $O(n^3)$ complexity in the size of the matrix. Since we do this for all possible subsets, the computational complexity to obtain $\phi_i(f,x)$ is $O(|N|^3 2^{|N|-1})$. To obtain $\phi_i(f,x)\forall i$, re-running this algorithm would result in a complexity of $O(|N|^4 2^{|N|-1})$. Alternatively, if we re-use terms in the summation we get a complexity of $O(|N|^3 2^{|N|})$.

Finally, to obtain $\phi_i(f,x)\forall i$ for M samples, we have to incur this exponential cost M

[1]The key observation is the for a linear $f(x)$, the expectation (and the conditional expectation) has the following property $\mathbb{E}[f(X)] = f(\mathbb{E}[\mathbb{X}])$

[2]In words, the conditional expectation for a normal distributed random variable is known to be $P_{\bar{S}}\mu + P_{\bar{S}}\Sigma P_S^T(P_S\Sigma P_S^T)^{-1}(P_S x - P_S\mu)$; however, this gives us a vector in $\mathbb{R}^{|S|}$. Since $f(x)$ expects an input in $\mathbb{R}^{|N|}$, we project the conditional expectation into $\mathbb{R}^{|N|}$ by multiplying by $P_{\bar{S}}$, where $\bar{S} \equiv N \setminus S$. The resulting vector has all of the features not in S set to zero. These features can simply be set to their known values since we are conditioning on them, hence the addition of $xP_S^T P_S$.

times for each explanation. Instead, we can isolate the exponential computation to a matrix that does not depend on x itself. This implies that if we can incur an exponential cost once, we can explain all samples in low order polynomial time. To do so, we can factor (Equation (3.7)) to get:

$$\mathbb{E}[x \mid x_S] = [Q_{\bar{S}} - U_S]\mu + [Q_S + U_S]x, \tag{3.8}$$

where $U_S = P_{\bar{S}}^T P_{\bar{S}} \Sigma P_S^T (P_S \Sigma P_S^T)^{-1} P_S$ and $Q_S = P_S^T P_S$. Then, if we use equation (Equation (3.8)) to revisit (Equation (3.5)), we get:

$$\phi_i(f, x) = \beta T^{(\mu)} \mu + \beta T^{(x)} x, \tag{3.9}$$

where $T^{(\mu)} = \frac{1}{M!} \sum_R ([Q_{\bar{S}^{R \cup i}} - U_{S^{R \cup i}}] - [Q_{\bar{S}^R} - U_{S^R}])$ and $T^{(x)} = \frac{1}{M!} \sum_R ([Q_{S^{R \cup i}} + U_{S^{R \cup i}}] - [Q_{S^R} + U_{S^R}])$. Here, we can see that computing $T^{(\mu)}$ and $T^{(x)}$ is exponential [3], however, once we have computed $T^{(\mu)}$ and $T^{(x)}$, we can compute the Shapley value $\phi_i(f, x)$ quickly[4].

Note that just as the original Shapley values have been approximated using Monte Carlo sampling, we can likewise approximate $T^{(\mu)}$ and $T^{(x)}$ by sampling from the permutations (or combinations) of feature orderings. In contrast to traditional sampling approaches which approximate the summation in Equation (3.5), approximating $T^{(\mu)}$ and $T^{(x)}$ converges much faster because we do not need to separately converge for each input feature.

3.3 Effects of Correlation

3.3.1 Impact of correlation on convergence

In order to build intuition about the conditional Shapley values for linear models, we first examine a simulated example with a known distribution. In the following example, the features are $x \in \mathbb{R}^3 \sim \mathcal{N}(0, \Sigma)$, the model is $f(x) = 1 \times x_1 + 2 \times x_2 + 3 \times x_3$, and the sample

[3] In fact, the complexity to compute them for all features is $O(|N|^3 2^{|N|})$

[4] In a few matrix multiplications and an addition.

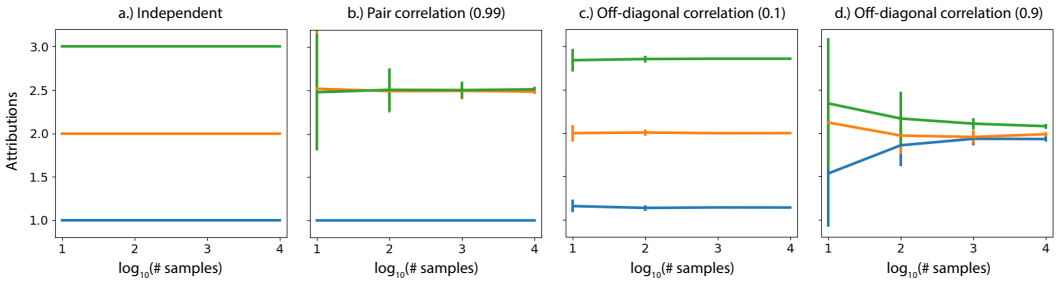


Figure 3.1: Convergence of correlated linear Shapley value estimates. x_1's attributions are blue, x_2 is orange, and x_3 is green. We report the standard deviations from 20 estimates of the conditional Shapley values using a fixed number of samples of combinations S.

being explained is $x^f = [1, 1, 1]$.

In Figure 3.1, there are three cases: (1) Independent implies that the correlation is the identity matrix $\Sigma = I$, (2) Pair correlation (ρ) implies that $\Sigma = I$, except $\Sigma_{2,3} = \Sigma_{3,2} = \rho$, and (3) Off-diagonal correlation (ρ) implies that $\Sigma_{i,j} = 1$ if $i = j$ and $\Sigma_{i,j} = \rho$ otherwise.

We can observe that when features are independent, the conditional Shapley value estimates are $\phi(x^f, f) = [1, 2, 3]$ which coincides with the marginal Shapley value estimates. Furthermore, for data that is truly independent, there is no variance in the estimates and they converge immediately. For other correlation patterns, we can observe two trends: 1.) correlation splits the β as credit between correlated variables and 2.) higher levels of correlation leads to slower convergence of the conditional Shapley value estimates.

3.3.2 Explaining a feature not used by the model

In order to compare marginal/conditional Shapley values on real data we utilize data from the National Health and Nutrition Examination Survey (NHANES). In particular, we focus on the task of predicting 5-year mortality within individuals (n=25,535) from 1999-2014, where mortality status is collected in 2015[5]. Note that conditional Shapley values require the covariance and mean of the underlying distribution, for which we use the sampling

[5]We filter out individuals with unknown mortality status.

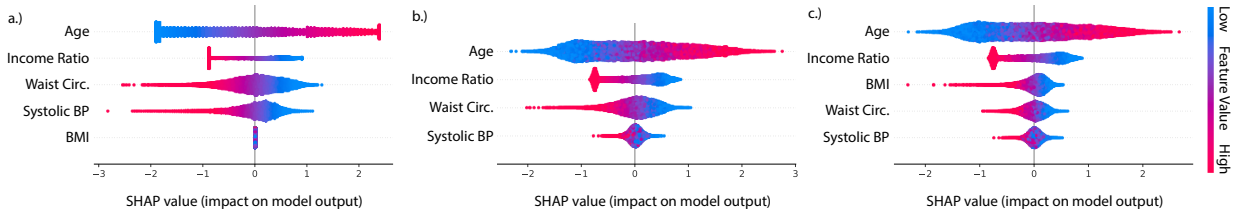


Figure 3.2: Marginal vs. conditional Shapley values for NHANES. In a.), we use the marginal approach whereas in b.) and c.), we use the conditional approach with different sets of features. In these summary plots, each point is an individual where the x-axis is the Shapley value, or the impact on the model's output). The color is the value of the feature listed in the y-axis.

covariance and mean.

For Figure 3.2, we use five features: *Age*, *Income ratio*, *Systolic blood pressure*, *Waist circumference*, and *Body mass index (BMI)*. In particular, we train a linear model on the first four features (excluding *BMI* (test AUC: 0.772, test AP: 0.186). Then, the marginal Shapley values give credit depending on the corresponding coefficient. In particular, *Age* positively impacts mortality prediction whereas *Income ratio*, *Waist circumference*, and *Systolic blood pressure* all negatively impact mortality prediction. Finally, *BMI* has no importance because it is not used in the model.

However, for the conditional Shapley values, we first observe that the number of features used to explain the model impact the attributions (four features in Figure 3.2b and five features in Figure 3.2c). In Figure 3.2b, we see that the relationships are similar, but slightly different to the marginal Shapley values in Figure 3.2a due to correlation in the data. In Figure 3.2c, we can see that when we include *BMI*, the importance of the other features is relatively lower. This implies that even though *BMI* is not included in the model, the correlation between *BMI* and other features makes BMI important under conditional Shapley values. Being able to explain features not in the model has implications for detecting bias. For instance, a linear model may use correlations between features to implicitly depend on a sensitive feature that was explicitly excluded. Conditional Shapley values provide a tool to identify such bias (though we note that in this case there are other approaches to identify

surrogate variables such as correlation analysis).

3.4 True to the Model or True to the Data

We now consider two examples using real world datasets and use cases that demonstrate why neither the conditional nor the marginal expectation are the right choice *in general*, but can be chosen based on the desired application. We use these applications to argue that the choice of conditional expectation comes down to whether you want your attributions to be *true to the model* or *true to the data*.

3.4.1 True to the Model

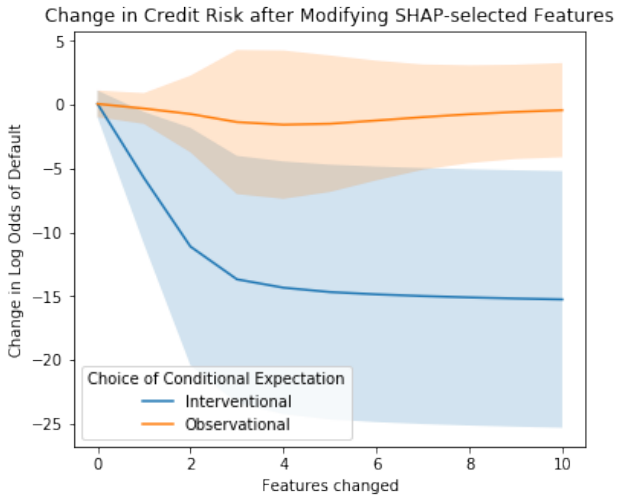


Figure 3.3: Modifying features according to the marginal Shapley values helps applicants decrease their predicted log odds of default much more than conditional Shapley values. Solid line indicates mean change in log odds, while shaded region indicates standard deviation over all applicants. The wide range is expected as applicants who are very close to the mean or with very low odds of default to begin with will not be able to further decrease their odds of default by setting features to the mean.

We first consider the case of a bank that uses an algorithm to determine whether or not to grant loans to applicants. Applicants who have been denied loans may want an explanation for why they were denied, and to understand what they would have to change to receive a

loan [19]. In this case, the mechanism we want to explain is the particular model the bank uses. In that case, we argue that we want our feature attributions to be *true to the model*. Therefore, we hypothesize that the marginal expectation is preferable, as it is the choice of value function that satisfies the Dummy axiom – only features that are referenced by the model will be given importance.

To investigate this, we downloaded the LendingClub dataset[6], a large dataset of loans issued from a peer-to-peer lending site which includes loan status and latest payment information, as well as a variety of features describing the applicants such as number of open bank accounts, age, and amount requested. We trained a logistic regression model to predict whether or not an applicant would default on their loan. We obtained feature attributions using either the conditional or marginal expectation.

To see which set of explanations was more useful to hypothetical applicants, we wanted to see which set of explanations helped applicants most decrease their risk of default according to the model (and consequently most increase their likelihood of being granted a loan). We therefore ranked all of the features for each applicant by their Shapley value, and allowed each applicant to "modify their risk" by setting that feature to the mean. We then measured the change in the model's predicted log odds of default after each feature (up to 10 features) had been mean-imputed. For this metric, the better the explanation, the faster the predicted log odds of default will decrease.

We find that using the marginal expectation leads to significantly better results than the conditional expectation (Figure 3.3). Intervening on the features ranked by the marginal Shapley values lead to a far greater decrease in predicted likelihood of default. In other words, the marginal Shapley values enabled interventions on individuals' features that drastically changed their predicted likelihood of receiving a loan.

When we consider the axioms fulfilled by each choice of value function, this result makes sense. As pointed out in Janzing et al. [87] and Sundararajan and Najmi [188], and as shown

[6]https://www.kaggle.com/wendykan/lending-club-loan-data

in subsection 3.3.2, the conditional Shapley value spreads importance among correlated features that may not be explicitly used by the machine learning model. Intervening on such features *will not* impact the model's output. In contrast, the marginal Shapley value is true to the model in the sense that it gives importance to features explicitly used by the model. For a linear model, this means the marginal approach will first change a feature i where $|\beta_i(x_i - \mu_i)|$ is largest. Compared to other features, mean imputing x_i will provide the greatest change to the predicted output of a linear model. Being true to the model is the best choice for most applications of explainable AI, where the goal is to explain the model itself.

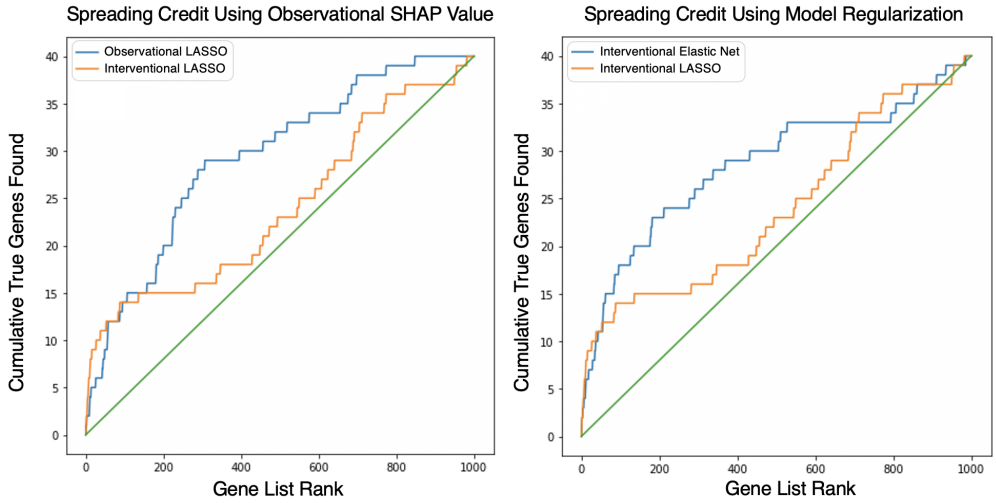


Figure 3.4: Left: When explaining a sparse model (Lasso regression), more true features are recovered when using the conditional Shapley value to spread credit among correlated features than using the marginal Shapley value. Right: When using the marginal Shapley value, we recover more true features when the underlying model spreads credit among groups of correlated features (Elastic Net) than when the underlying model is sparse (Lasso).

3.4.2 True to the Data

We now consider the complementary case where we care less about the particular machine learning model we have trained, and more about a natural mechanism in the world. We use a

dataset of RNA-seq gene expression measurements in patients with acute myeloid leukemia, a blood cancer [195]. An important problem in cancer biology is to determine which genes' expression determine a particular outcome (e.g., response to anti-cancer drugs). One common approach is to measure gene expression in a set of patient samples, measure response to drugs *in vitro*, then use machine learning to model the data and examine the weights of the model [217, 113, 86].

To create an experimental setting where we have access to the ground truth, we take the real RNA-seq data and simulate a drug response label as a function of 40 randomly selected *causal genes* (out of 1000 total genes). The label is defined to be the sum of the causal genes plus gaussian noise. After training a Lasso model, we explain the model for many samples using the conditional and marginal Shapley values and rank the genes by their average magnitude Shapley value to get two sets of global feature importance values [123]. We see that ranking the features according to the conditional Shapley values recovers more of the true causal features at each position in the ranked list than the marginal Shapley values (Figure 3.4, left). The green line in the figure represents the expected number of true genes that would be cumulatively found at each position in the ranked list if the gene list were sorted randomly. While we see that both Shapley value-based rankings outperform random rankings, the conditional approach outperforms the marginal one.

This example helps to illustrate why the Dummy axiom, while important for *model explanation*, is not necessarily a useful axiom in general. In the case of biological discovery, we do not care about the particular linear model we have trained on the data. We instead care about the true data generating process, which may be equally well-represented by a wide variety of models [24]. Therefore, when ranking genes for further testing, we want to spread credit among correlated features that are all informative about the outcome of interest, rather than assigning no credit to features that are not explicitly used by a single model.

While the conditional Shapley value may be preferable to the marginal Shapley value when explaining a Lasso model, Elastic Net (i.e., a penalty on L1 and L2 norm of the

coefficients) is actually more popular for this application [217]. While a Lasso model may achieve high predictive performance, it will attempt to sparsely pick out features from among groups of correlated features. We re-run the same experiment, but rather than comparing the conditional and marginal Shapley values applied to a Lasso model, we focus on the marginal Shapley values for (1) a Lasso model (as in the previous experiment), or (2) an Elastic Net model (Figure 3.4, right). We find that by using the Elastic Net regularization penalty, the *model* itself is able to spread credit among correlated features, better respecting the correlation in the data. It is worthwhile to point out here that Elastic Net models became popular for this task because it is typical practice to interpret linear models by examining the coefficient vector, which is itself an marginal style explanation (partial derivative). Since this marginal explanation does not spread credit among correlated features, it is necessary to spread the credit using modeling decisions.

We have seen that when the goal is to be *true to the data*, there are two methods for spreading credit to correlated features. One is to spread credit using the conditional Shapley value as a feature attribution. The other is to train a model that itself spreads credit among correlated features, in this case by training an Elastic Net regression. When we factor in the computation time for these two approaches, the choice becomes clear. Estimating the transform matrices for the conditional with 1000 samples took 6 hours using the CPUs on a 2018 MacBook Pro, while hyperparameter tuning and fitting an Elastic Net regression took a matter of seconds.

3.5 Discussion

In this paper, we analyzed two approaches to explain models using the Shapley value solution concept for cooperative games. In order to compare these approaches we focus on explaining linear models and present a novel methodology for explaining linear models with correlated features. We analyze two different settings where either the marginal Shapley values or the conditional Shapley values are preferable. In the first setting, we consider a model trained on loans data that might be used to determine which applicants obtain loans. Because

applicants in this setting are ultimately interested in why the model makes a prediction, we call this case "true to the model" and show that marginal Shapley values serve to modify the model's prediction more effectively. In the second setting we consider a model trained on biological data that aims to understand an underlying causal relationship. Because this setting is focused on scientific discovery, we call this case "true to the data" and show that for a sparse model (Lasso regularized) conditional Shapley values discover more of the true features. We also find that modeling decisions can achieve some of the same effects, by demonstrating that the marginal Shapley values recover more of the true features when applied to a model that itself spreads credit among correlated features than when applied to a sparse model.

Limitations and future directions: In the RNA-seq experiment we identified two solutions to identify true features: (1) Lasso regression with conditional Shapley values where correlation is spread through the attribution method and (2) Elastic Net regression with marginal Shapley values where correlation is spread through the model estimation. While both approaches achieved similar efficacy, we found that the latter was far more computationally tractable. As future work, we aim to further analyze which of these approaches are preferable or even feasible for scientific discovery beyond linear models.

Currently, the best case for feature attribution is when the features that being perturbed are independent to start with. In that case, both the marginal and conditional approaches yield the same attributions. Therefore, future work that focuses on reparameterizing the model to get at the underlying independent factors is a promising approach to eliminate the *true to the model* vs. *true to the data* tradeoff, where we can perturb the data interventionally without generating unrealistic input values.

Chapter 4

EXPLAINING TREE MODELS

4.1 Introduction

Machine learning models based on trees are the most popular non-linear models in use today [94, 65]. Random forests, gradient boosted trees, and other tree-based models are used in finance, medicine, biology, customer retention, advertising, supply chain management, manufacturing, public health, and other areas to make predictions based on sets of input features (Figure 4.1A left). For these applications, models often must be *both accurate and interpretable*, where interpretability means that we can understand how the model uses input features to make predictions [121]. However, despite the rich history of *global* interpretation methods for trees, which summarize the impact of input features on the model as a whole, much less attention has been paid to *local* explanations, which reveal the impact of input features on individual predictions (i.e., for a single sample) (Figure 4.1A right).

Current local explanation methods include: 1) reporting the decision path, 2) using a heuristic approach that assigns credit to each input feature [167], and 3) applying various model-agnostic approaches that require repeatedly executing the model for each explanation [160, 47, 184, 121, 13]. Each current method has limitations. First, simply reporting a prediction's decision path is unhelpful for most models, particularly those based on multiple trees. Second, the behavior of the heuristic credit allocation has yet to be carefully analyzed; we show here that it is strongly biased to alter the impact of features based on their tree depth. Third, since model-agnostic methods rely on post hoc modeling of an arbitrary function, they can be slow and suffer from sampling variability.

We present TreeExplainer, an explanation method for trees that enables the tractable computation of *optimal* local explanations, as defined by desirable properties from game

theory. TreeExplainer bridges theory to practice by building on previous model-agnostic work based on classic game-theoretic Shapley values [121, 172, 184, 47, 188, 87]. It makes three notable improvements.

1. *Exact computation of Shapley value explanations for tree-based models.* Classic Shapley values can be considered "optimal" since, within a large class of approaches, they are the only way to measure feature importance while maintaining several natural properties from cooperative game theory [121, 87]. Unfortunately, in general these values can only be approximated since computing them exactly is NP-hard [130], requiring a summation over all feature subsets. Sampling-based approximations have been proposed [184, 121, 47]; however, using them to compute low variance versions of the results in this paper for even our smallest dataset would consume years of CPU time (particularly for interaction effects). By focusing specifically on trees, we developed an algorithm that computes local explanations based on exact Shapley values in polynomial time. This provides local explanations with *theoretical guarantees* of *local accuracy* and *consistency* [121] (Methods).

2. *Extending local explanations to directly capture feature interactions.* Local explanations that assign a single number to each input feature, while very intuitive, cannot directly represent *interaction* effects. We provide a theoretically grounded way to measure local interaction effects based on a generalization of Shapley values proposed in game theory literature [68]. We show that this approach provides valuable insights into a model's behavior.

3. *Tools for interpreting global model structure based on many local explanations.* The ability to efficiently and exactly compute local explanations using Shapley values across an entire dataset enables the development of a range of tools to interpret a model's global behavior (Figure 4.1B). We show that combining many local explanations lets us represent global structure while retaining *local faithfulness* [161] to the original model,

which produces detailed and accurate representations of model behavior.

Explaining predictions from tree models is particularly important in medical applications, where the patterns a model uncovers can be more important than the model's prediction performance [174, 122]. To demonstrate TreeExplainer's value, we use three medical datasets, which represent three types of loss functions: 1) *Mortality*, a dataset with 14,407 individuals and 79 features based on the NHANES I Epidemiologic Followup Study [45], where we model the risk of death over twenty years of followup. 2) *Chronic kidney disease*, a dataset that follows 3,939 chronic kidney disease patients from the Chronic Renal Insufficiency Cohort study over 10,745 visits, where we use 333 features to classify whether patients will progress to end-stage renal disease within 4 years. 3) *Hospital procedure duration*, an electronic medical record dataset with 147,000 procedures and 2,185 features, where we predict duration of a patient's hospital stay for an upcoming procedure.

In this paper, we discuss how the accuracy and interpretability of tree-based models make them appropriate for many applications. We then describe why these models need more precise local explanations and how we address that need with TreeExplainer. Next, we extend local explanations to capture interaction effects. Finally, we demonstrate the value of explainable AI tools that combine many local explanations from TreeExplainer (https://github.com/suinleelab/treeexplainer-study).

4.2 Advantages of tree-based models

Tree-based models can be more accurate than neural networks in many applications. While deep learning models are more appropriate in fields like image recognition, speech recognition, and natural language processing, tree-based models consistently outperform standard deep models on tabular-style datasets, where features are individually meaningful and lack strong multi-scale temporal or spatial structures [35]. The three medical datasets we examine here all represent tabular-style data. Gradient boosted trees outperform both pure deep learning and linear regression across all three datasets (Figure 4.3A).

Tree-based models can also be more interpretable than linear models due to model-mismatch effects. It is well-known that the bias/variance trade-off in machine learning has implications for model accuracy. Less well appreciated is that this trade-off also affects interpretability. Simple high-bias models (such as linear models) seem easy to understand, but they are sensitive to model mismatch, i.e., where the model's form does not match its true relationships in the data [78]. This mismatch can create hard-to-interpret model artifacts.

To illustrate why low-bias models can be more interpretable than high-bias ones, we compare gradient boosted trees to linear logistic regression using the mortality dataset. We simulate a binary outcome based on a participant's age and body mass index (BMI), and we vary the amount of non-linearity in the simulated relationship (Figure 4.3B). As expected, by increasing non-linearity, the bias of the linear model causes accuracy to decline (Figure 4.3C). Perhaps unexpectedly, it also causes interpretability to decline (Figure 4.3D). We know that the model should depend only on age and BMI, but even a moderate amount of non-linearity in the true relationship causes the linear model to begin using other irrelevant features (Figure 4.3D), and the weight placed on these features is driven by complex cancellation effects that are not readily interpretable. When a linear model depends on cancellation effects between irrelevant features, the function itself is not complicated, but the meaning of the features it depends on become subtle: they are no longer being used primarily for their marginal effects, but rather for their interaction effects. Thus, even when simpler high-bias models achieve high accuracy, low-bias ones may be preferable, and even more interpretable, since they are likely to better represent the true data-generating mechanism and depend more naturally on their input features.

4.3 Tree SHAP

As previously mentioned, Shapley values have been adapted to explain machine learning algorithms; however, calculating them is NP-hard. By restricting our machine learning model class to trees, we create tractable algorithms to exactly estimate marginal Shapley values and approximately estimate conditional Shapley values.

In this thesis, we will focus on describing the algorithm to estimate marginal Shapley values, because it is exact[1]. As previously mentioned, marginal Shapley values are equivalent to the average of baseline Shapley values with many baselines. Moving forward, our goal is to design a tractable algorithm to compute baseline Shapley values.

The brute force approach to estimate baseline Shapley values for a single feature ($\phi_i(f, x^e, x^b)$) involves making predictions to compute marginal contributions for each possible subset. If we assume the computational cost of computing the weight is constant[2], then the complexity of the brute force method is the number of terms in the summation multiplied by the cost of making a prediction (on the order of the depth of the tree). This gives $O((\text{tree depth}) \times 2^d)$, where d is the number of features (analogous to the number of players $|D|$).

Then, in order to compute $\phi_i(f, x^e, x^b)$ for all features i, we have to re-run the entire algorithm d times, giving us an overall complexity of

$$O(d \times (\text{tree depth}) \times 2^d) \tag{4.1}$$

An exponential computational complexity is bad; however, if we *constrain* $f(x)$ *to be a tree-based model* (e.g., XGBoost, decision trees, random forests, etc.), then we can come up with a polynomial time algorithm to compute $\phi_i(f, x^e, x^b)$ exactly. Why is this the case? Well, even for explaining a single feature, the brute force algorithm may consider a particular path multiple times. However, to compute the Baseline Shapley value for a single feature, it turns out that we only need to consider each path once (Theorem 1).

Theorem 1. *To calculate $\phi_i(f, x^f, x^b)$, we can calculate attributions for each path from the root to each leaf. For a given path P, we define N_P to be the unique features in the path and S_P to be the unique features in the path that came from x^f. Finally, define v to be the value*

[1]This algorithm is my (Hugh Chen) contribution to the paper.

[2]Which it will be if we memoize $k!$ for $k = 1, \cdots, |D|$.

of the path's leaf. Then, the attribution of the path is:

$$\phi_i^P(f, x^f, x^b) = \begin{cases} 0 & \text{if } i \notin N_P \\ W(|S_P| - 1, |N_P|) \times v & \text{if } i \in S_P \\ -W(|S_P|, |N_P|) \times v & \text{o.w.} \end{cases} \tag{4.2}$$

Proof. (Sketch) Treat each path P in the tree from the root to each leaf as a separate model $f^P(x)$ that returns the value of the leaf if that path is traversed by x or zero otherwise. This results in (# leaf nodes) models that operate on disjoint parts of the input space, implying that our original model is equal to the summation of all of these path models:

$$f(x) = \sum_P f^P(x) \tag{4.3}$$

By the additivity of Shapley values,

$$\phi_i(f, x^f, x^b) = \sum_P \phi_i(f^P, x^f, x^b) \tag{4.4}$$

Then, we can simply calculate ϕ_i for each path model. Since the path model is zero everywhere except for the associated path, we arrive to the solution in Theorem 1. □

Then, we can design an algorithm which keeps track of features for each path as we perform a traversal of the tree model. The resultant algorithm checks against a constantly updating list of current features and calculates the marginal Shapley values with the following computational complexity:

$$O((\#\text{ internal nodes}) \times (\text{tree depth})) + O((\#\text{ leaf nodes}) \times (\text{tree depth}) \times d)$$

We can first get rid of the multiplicative (tree depth) factor by using binary arrays to indicate the presence of features in a given path instead of lists (data structures). Next,

we can get rid of the final multiplicative (d) factor by computing the attributions for all features simultaneously as we traverse the tree by passing "Pos" and "Neg" attributions to parent nodes (dynamic programming). We can first observe that for each leaf, according to Theorem 1, there are only two possible values needed to compute the Baseline Shapley value (Pos and Neg). Then, in Figure 4.2, based on the attributions for x_1 we see that these Pos and Neg terms can be grouped by the left and right subtrees below x_1. To generalize, we make the following observation:

Observation *In order to compute the attribution for any feature i it is sufficient to consider the paths that correspond to each internal node's children. Then, we can focus on internal n, where we know that one child is associated with x^f child and one child is associated with x^b. The attribution to the node's feature is:*

$$\sum_{\text{paths } P \text{ under } x^f \text{ child}} \text{Pos}_P + \sum_{\text{paths } P \text{ under } x^b \text{ child}} \text{Neg}_P \tag{4.5}$$

The final algorithm, which has a computational complexity of $O(\#nodes)$ is:

```
def dynamic(array xf, array xb, tree T):
    phi = [0]*len(xf)
    def recurse(node n, int nc, int sc, array fseen, array bseen):
        # Case 1: Leaf
        if n.is_leaf:
            if sc == 0: return((0,0))
            else: return((n.value*W(sc,nc-1),-n.value*W(sc,nc)))
        # Find children associated with xf and xb
        xf_child = n.left if xf[n.feat] < n.thres else n.right
        xb_child = n.left if xb[n.feat] < n.thres else n.right
        # Case 2: Feature encountered before
        if fseen[n.feat] > 0:
            return(recurse(xf_child,nc,sc,fseen,bseen))
```

```
    if bseen[n.feat] > 0:
        return(recurse(xb_child,nc,sc,fseen,bseen))
    # Case 3: xf and xb go the same way
    if xf_child == xb_child:
        return(recurse(xb_child,nc,sc,fseen,bseen))
    # Case 4: xf and xb don't go the same way
    if xf_child != xb_child:
        fseen[n.feat] += 1
        posf,negf = recurse(xf_child,nc+1,sc+1,fseen,bseen)
        fseen[n.feat] -= 1; bseen[n.feat] += 1
        posb,negb = recurse(xb_child,nc+1,sc  ,fseen,bseen)
        bseen[n.feat] -= 1
        phi[n.feat] += posf+negb
        return((posf+posb,negf+negb))
    recurse(n=T.root,0,0,[0]*len(xf),[0]*len(xf))
```

4.4 Local explanations for trees

Current local explanations for tree-based models are *inconsistent*. To our knowledge, only two tree-specific approaches can quantify a feature's *local* importance for an individual prediction. The first is simply reporting the decision path, which is unhelpful for ensembles of many trees. The second is an unpublished heuristic approach (proposed by *Saabas* [167]), which explains a prediction by following the decision path and attributing changes in the model's expected output to each feature along the path. The Saabas method has not been well studied, and we demonstrate here it is biased to alter the impact of features based on their distance from a tree's root. This bias makes Saabas values *inconsistent*, where increasing a model's dependence on a feature may actually decrease that feature's Saabas value . This is the opposite of what an effective attribution method should do. We show this difference by

examining trees representing multi-way AND functions, for which no feature should have more credit than another. Yet Saabas values give splits near the root much less credit than those near the leaves. Consistency is critical for an explanation method since it makes comparisons among feature importance values meaningful.

Model-agnostic local explanation approaches are slow and variable. While model-agnostic local explanation approaches can explain tree models, they rely on post hoc modeling of an arbitrary function and thus can be slow and/or suffer from sampling variability when applied to models with many input features. To illustrate this, we generate random datasets of increasing size and then explain (over)fit XGBoost models with 1,000 trees. This runtime of this experiment shows a linear increase in complexity as the number of features increases; model-agnostic methods take a significant amount of time to run over these datasets, even though we allowed for non-trivial estimate variability and used only a moderate number of features. While often practical for individual explanations, model-agnostic methods can quickly become impractical for explaining entire datasets.

TreeExplainer provides fast local explanations with guaranteed consistency. It bridges theory to practice by reducing the complexity of exact Shapley value computation from exponential to polynomial time. This is important since within the class of *additive feature attribution methods*, a class that we have shown contains many previous approaches to local feature attribution [121], results from game theory imply the Shapley values are the only way to satisfy three important properties: *local accuracy*, *consistency*, and *missingness*. Local accuracy (known as *efficiency* in game theory) states that when approximating the original model f for a specific input x, the explanation's attribution values should sum up to the output $f(x)$. Consistency (known as *monotonicity* in game theory) states that if a model changes so that some feature's contribution increases or stays the same regardless of the other inputs, that input's attribution should not decrease. Missingness (*null effects* and *symmetry* in game theory), is a trivial property satisfied by all previous explanation methods.

TreeExplainer enables the *exact* computation of Shapley values in *low order polynomial time* by leveraging the internal structure of tree-based models. Shapley values require a sum-

mation of terms over all possible feature subsets, TreeExplainer collapses this summation into a set of calculations specific to each leaf in a tree (Methods). This represents an exponential complexity improvement over previous exact Shapley methods. To compute the impact of a specific feature subset during the Shapley value calculation, TreeExplainer uses marginal expectations over a user-supplied background dataset [87]. But it can also avoid the need for a user-supplied background dataset by relying only on the path coverage information stored in the model (which is usually from the training dataset).

Efficiently and exactly computing the Shapley values guarantees that explanations are always consistent and locally accurate, improving results over previous local explanation methods in several ways:

- *Impartial feature credit assignment regardless of tree depth.* In contrast to Saabas values, Shapley values allocate credit uniformly among all features participating in multi-way AND operations and thereby avoid inconsistency problems.

- *No estimation variability.* Since solutions from model-agnostic sampling methods are approximate, TreeExplainer's exact explanations eliminate the additional burden of checking their convergence and accepting a certain amount of noise in the estimates (other than noise from the choice of a background dataset).

- *Strong benchmark performance* (Figure 4.4). We designed 15 metrics to comprehensively evaluate the performance of local explanation methods; we applied these metrics to ten different explanation methods across three different model types and three datasets. Results for the chronic kidney disease dataset, shown in Figure 4.4, demonstrate consistent performance improvements for TreeExplainer.

- *Consistency with human intuition.* We evaluated how well explanation methods match human intuition by comparing their outputs with human consensus explanations of 12 scenarios based on simple models. Unlike the heuristic Saabas values, Shapley-value-based explanation methods agree with human intuition in all tested scenarios.

TreeExplainer also extends local explanations to measure interaction effects. Traditionally, local explanations based on feature attribution assign a single number to each input feature. The simplicity of this natural representation comes at the cost of conflating main and interaction effects. While interaction effects between features can be reflected in the global patterns of many local explanations, their distinction from main effects is lost in each local explanation (Figure 4.5B-G).

We propose *SHAP interaction values* as a richer type of local explanation. These values use the 'Shapley interaction index' from game theory to capture local interaction effects. They follow from generalizations of the original Shapley value properties [68] and allocate credit not just among each player of a game, but among all pairs of players. The SHAP interaction values consist of a matrix of feature attributions (interaction effects on the off-diagonal and the remaining effects on the diagonal). By enabling the separate consideration of interaction effects for individual model predictions, TreeExplainer can uncover significant patterns that might otherwise be missed.

4.5 Local explanations as building blocks for global insights

Previous approaches to understanding a model globally focused on using simple global approximations [65], finding new interpretable features [99], or quantifying the impact of specific internal nodes in a deep network [212, 16, 114]. We present methods that combine many local explanations to provide global insight into a model's behavior. This lets us retain local faithfulness to the model while still capturing global patterns, resulting in richer, more accurate representations of the model's behavior.

Local model summarization reveals rare high-magnitude effects on mortality risk and increases feature selection power. Combining local explanations from TreeExplainer across an entire dataset enhances traditional global representations of feature importance by: (1) avoiding the inconsistency problems of current methods, (2) increasing the power to detect true feature dependencies in a dataset, and (3) enabling us to build *SHAP summary plots*, which succinctly display the magnitude, prevalence, and direction of a feature's effect. SHAP

summary plots avoid conflating the magnitude and prevalence of an effect into a single number, and so reveal rare high magnitude effects. Figure 4.5A (right) reveals the direction of effects, such as men (blue) having a higher mortality risk than women (red); and the distribution of effect sizes, such as the long right tails of many medical test values. These long tails mean features with a low global importance can be extremely important for specific individuals. Interestingly, rare mortality effects always stretch to the right, which implies there are many ways to die abnormally early when medical measurements are out-of-range, but not many ways to live abnormally longer.

Local feature dependence reveals both global patterns and individual variability in mortality risk and chronic kidney disease. *SHAP dependence plots* show how a feature's value (x-axis) impacts the prediction (y-axis) of every sample (each dot) in a dataset (Figure 4.5B and E). They provide richer information than traditional partial dependence plots. For the mortality model, SHAP dependence plots reproduce the standard risk inflection point of systolic blood pressure [75], while also highlighting that the impact of blood pressure on mortality risk differs for people of different ages (Figure 4.5B). These types of interaction effects show up as vertical dispersion in SHAP dependence plots.

For the chronic kidney disease model, a dependence plot again clearly reveals a risk inflection point for systolic blood pressure. However, in this dataset the vertical dispersion from interaction effects appears to be partially driven by differences in blood urea nitrogen (Figure 4.5E). Correctly modeling blood pressure risk while retaining interpretabilty is vital because blood pressure control in select chronic kidney disease (CKD) populations may delay progression of kidney disease and reduce the risk of cardiovascular events.

Local interactions reveal sex-specific life expectancy changes during aging as well as inflammation effects in chronic kidney disease. Using SHAP interaction values, we can decompose the impact of a feature on a specific sample into *interaction effects* with other features. This helps us measure global interaction strength as well as decompose SHAP dependence plots into interaction effects at a local (i.e., per sample) level (Figure 4.5B-D). In the mortality dataset, plotting the SHAP interaction value between age and sex shows a

clear change in the relative risk between men and women over a lifetime (Figure 4.5G). The largest difference in risk between men and women occurs at age 60; the increased risk for men could be driven by their increased cardiovascular mortality relative to women near that age [141]. This pattern is not clearly captured without SHAP interaction values because being male always confers greater risk of mortality than being female (Figure 4.5A).

In the chronic kidney disease model, we identify an interesting interaction (Figure 4.5F): high white blood cell counts are more concerning to the model when they are accompanied by high blood urea nitrogen. This supports the notion that inflammation may interact with high blood urea nitrogen to hasten kidney function decline [21, 60].

Local model monitoring reveals previously invisible problems with deployed machine learning models. Using TreeExplainer to explain a model's *loss*, instead of a model's prediction, can improve our ability to monitor deployed models. Monitoring models is challenging because of the many ways relationships between the input and model target can change post-deployment. Detecting when such changes occur is difficult, so many bugs in machine learning pipelines go undetected, even in core software at top tech companies [216]. We demonstrate that local model monitoring helps debug model deployments and identify problematic features (if any) directly by decomposing the loss among the model's input features.

We simulated a model deployment with the hospital procedure duration dataset using the first year of data for training and the next three years for deployment. We present three examples: one intentional error and two previously undiscovered problems. (1) We intentionally swapped the labels of operating rooms 6 and 13 partway through the deployment to mimic a typical feature pipeline bug. The overall loss of the model's prediction gives no indication of the bug (Figure 4.6A), whereas the *SHAP monitoring plot* for the room 6 feature clearly identifies the labeling error (Figure 4.6B). (2) Figure 4.6C shows a spike in error for the general anesthesia feature shortly after the deployment window begins. This spike corresponds to a subset of procedures affected by a previously undiscovered temporary electronic medical record configuration problem. (3) Figure 4.6D shows an example of feature drift over time, not of a processing error. During the training period and early

in deployment, using the 'atrial fibrillation' feature lowers the loss; however, the feature becomes gradually less useful over time and eventually degrades the model. We found this drift was caused by significant changes in atrial fibrillation ablation procedure duration driven by technology and staffing changes. Current deployment practice monitors both the overall loss of a model (Figure 4.6A) over time and potentially statistics about input features. Instead, TreeExplainer lets us directly monitor the impact individual features have on a model's loss.

Local explanation embeddings reveal population subgroups relevant to mortality risk and complementary diagnostic indicators in chronic kidney disease. Unsupervised clustering and dimensionality reduction are widely used to discover patterns characterizing subgroups of samples (e.g., study participants), such as disease subtypes [198, 179]. These techniques have two drawbacks: 1) the distance metric does not account for discrepancies among units/meaning of features (e.g., weight vs. age), and 2) an unsupervised approach cannot know which features are relevant for an outcome of interest and so should be weighted more strongly. We address both limitations using *local explanation embeddings* to embed each sample into a new "explanation space." Running clustering in this new space will yield a *supervised clustering*, where samples are grouped based on their explanations. Supervised clustering naturally accounts for the differing units of various features, highlighting only the changes relevant to a particular outcome.

Running hierarchical supervised clustering using the mortality model results in many groups of people that share a similar mortality risk for similar reasons (Figure 4.7A). This grouping of samples can reveal high level structure in datasets that would not be revealed using standard unsupervised clustering and has various applications, from customer segmentation, to model debugging, to disease sub-typing. Analogously, we can also run PCA on local explanation embeddings for chronic kidney disease samples. This uncovers two primary categories of risk factors that identify unique individuals at risk of end-stage renal disease: (1) factors based on urine measurements, and (2) factors based on blood measurements (Figure 4.7B-D). This pattern is notable because it continues as we plot more top features. The separation between blood and urine features is consistent with the fact that clinically these

factors should be measured in parallel. This type of insight into the overall structure of kidney risk is not at all apparent in a standard unsupervised embedding.

4.6 Discussion

The potential impact of local explanations for tree-based machine learning models is widespread. Explanations can help satisfy transparency requirements, facilitate human/AI collaboration, and aid model development, debugging, and monitoring.

Tree-based machine learning models are widely used in many regulated domains, such as healthcare, finance, and public services. Improved interpretability is vital for these applications. In healthcare, the unknowing deployment of "Clever Hans" predictors that depend on spurious correlations could lead to serious patient harm [110, 152]. In finance, consumer protection laws require explanations for credit decisions, and no accepted standard exists for how to produce these for complex tree-based models [83]. In public service applications, explainability can promote accountability and anti-discrimination policies [49].

Improving human/AI collaboration is critical for applications where explaining machine learning model predictions can enhance human performance. Such applications include predictive medicine, customer retention, and financial model supervision. Local explanations enable support agents to predict why the customer they are calling is likely to leave. They enable doctors to make more informed decisions rather than blindly trust an algorithm's output. With financial model supervision, local explanations help human experts understand why the model made a specific recommendation for high-risk decisions.

Improving model development, debugging, and monitoring leads to more accurate and reliable deployments of machine learning systems. Local explanations aid model development by revealing which features are most informative for specific subsets of samples. They aid debugging by revealing the global patterns of how a model depends on its input features, and so enable developers to determine when patterns are unlikely to generalize well. Finally, they aid model monitoring by enabling the allocation of global accuracy measures among each model input, significantly increasing the signal-to-noise ratio for detecting problematic

data distribution shifts.

In this paper, we identified ways to significantly enhance the interpretability of tree-based models and to broaden the application of local explanation methods. We develop the first polynomial-time algorithm to compute Shapley values for trees. This algorithm solves what is in general an NP-hard problem in polynomial time for an important class of value functions. We present a richer type of local explanation that directly captures interaction effects. We demonstrate how using local explanation methods to explain model loss enables a more sensitive and informative method of model monitoring. We offer many tools for model interpretation that combine local explanations, such as dependence plots, summary plots, supervised clusterings, and explanation embeddings. We demonstrate that Shapley-based local explanations can improve upon state-of-the-art feature selection for trees. We identify under-appreciated interpretability problems with simple linear models. And we compile many varied explainability metrics into a unified open source benchmark, on which TreeExplainer consistently outperforms other alternatives. Local explanations have a distinct advantage over global ones. By focusing only on a single sample, they remain more faithful to the original model. By designing efficient and trustworthy ways to obtain local explanations for modern tree-based models, we take an important step toward enabling local explanations to become foundational building blocks for an ever growing number of downstream machine learning tasks.

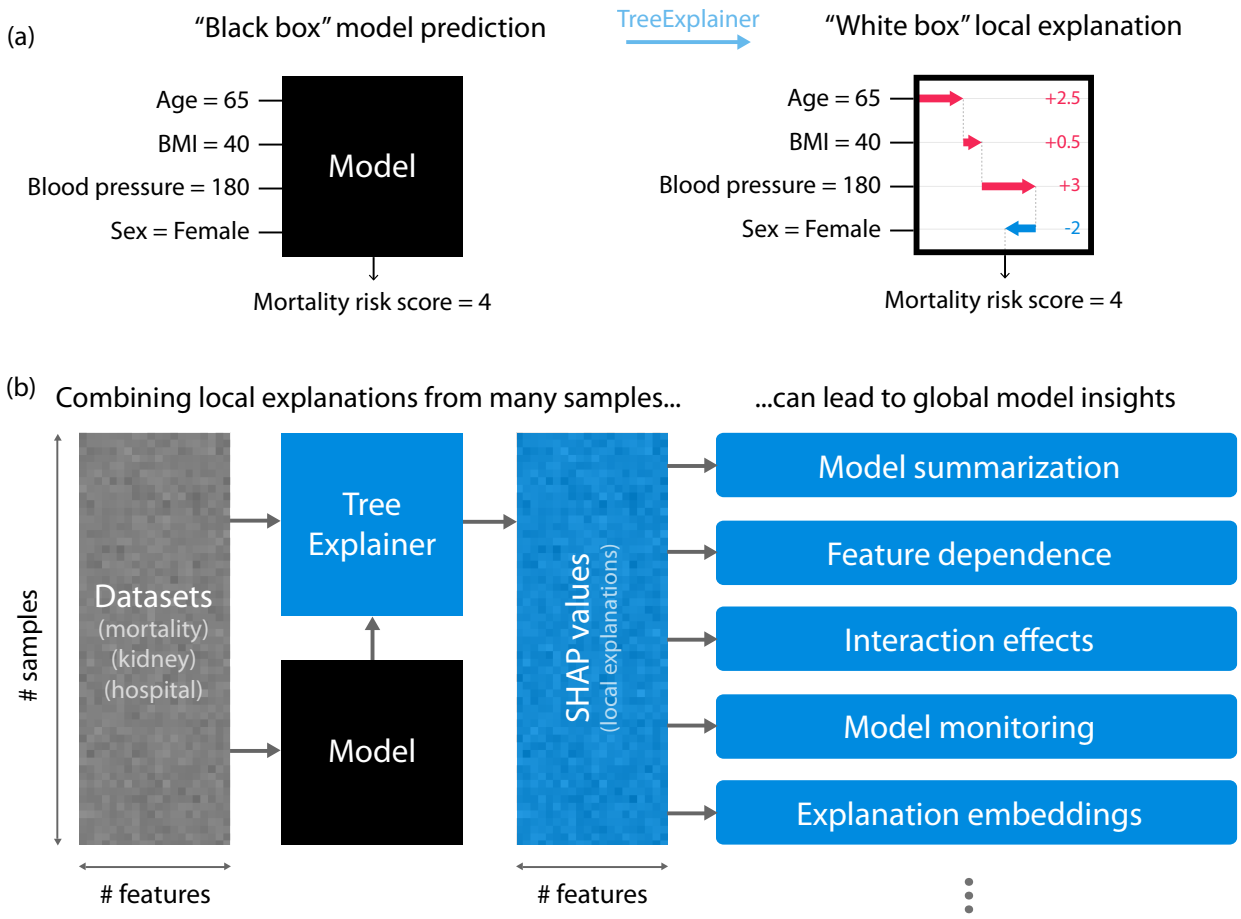
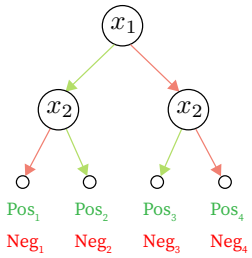


Figure 4.1: **Local explanations based on TreeExplainer enable a wide variety of new ways to understand global model structure.** (a) A local explanation based on assigning a numeric measure of credit to each input feature. (b) By combining many local explanations, we can represent global structure while retaining local faithfulness to the original model. We demonstrate this by using three medical datasets to train gradient boosted decision trees and then compute local explanations based on SHapley Additive exPlanation (SHAP) values [121]. Computing local explanations across all samples in a dataset enables development of many tools for understanding global model structure.



$\phi_1(f, x^f, x^b)$	$\text{Pos}_1 + \text{Pos}_2 + \text{Neg}_3 + \text{Neg}_4$
$\phi_2(f, x^f, x^b)$	$\text{Neg}_1 + \text{Pos}_2 + \text{Pos}_3 + \text{Neg}_4$

Figure 4.2: Example to illustrate collapsibility for features. Green paths are associated with x^f and red paths are x^b.

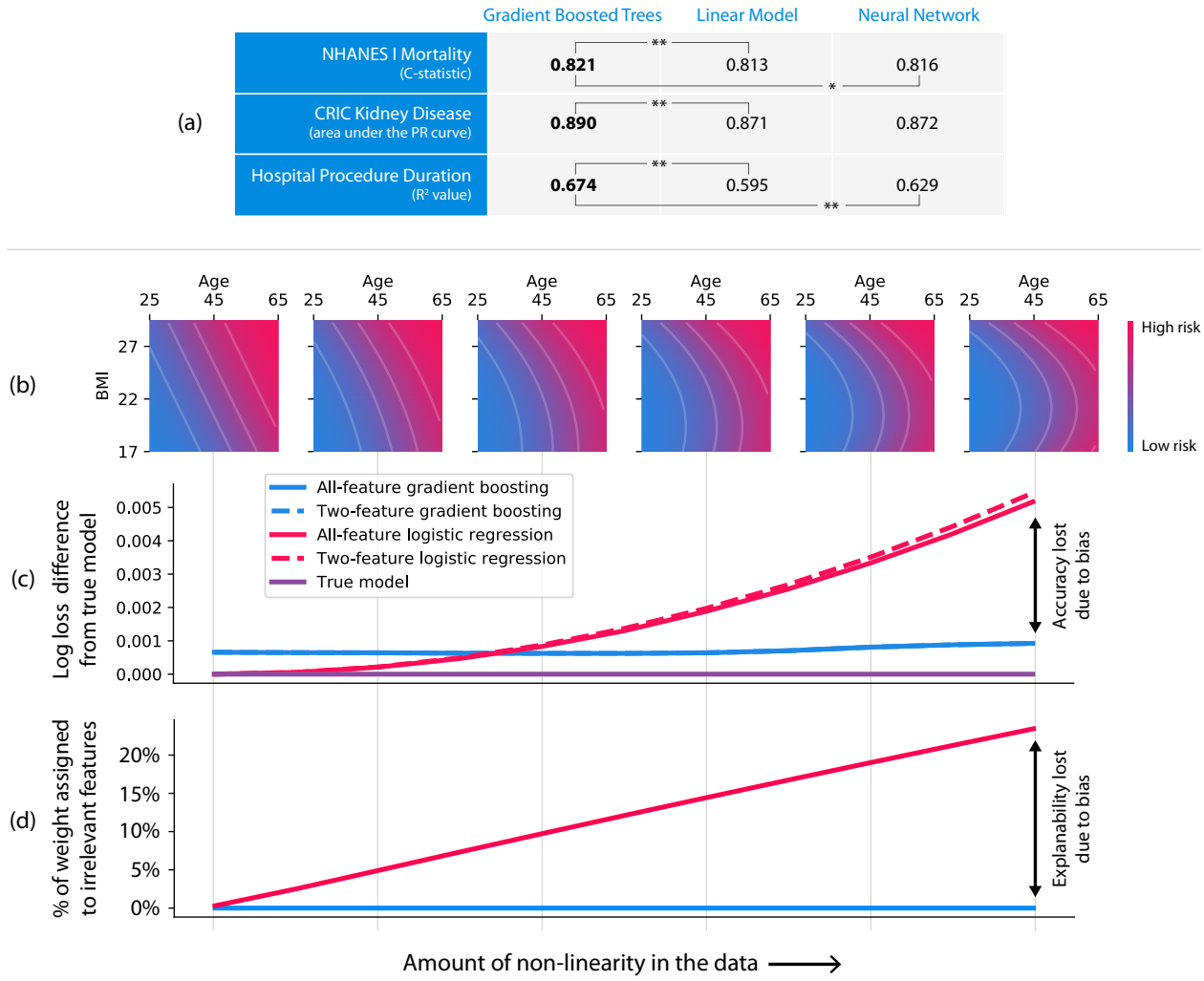


Figure 4.3: **Gradient boosted tree models can be more accurate than neural networks and more interpretable than linear models.** (a) Gradient boosted tree models outperform both linear models and neural networks on all our medical datasets, where (**) represents a bootstrap retrain P-value < 0.01, and (*) represents a P-value of 0.03. (b-d) Linear models exhibit explanation and accuracy error in the presence of non-linearity. (b) The data generating models we used for the simulation ranged from linear to quadratic along the body mass index (BMI) dimension. (c) Linear logistic regression (red) outperformed gradient boosting (blue) up to a specific amount of non-linearity. Not surprisingly, the linear model's bias is higher than the gradient boosting model's, as shown by the steeper slope as we increase non-linearity. (d) As the true function becomes more non-linear, the linear model assigns more credit (coefficient weight) to features that were not used by the data generating model.

Figure 4.4: **Explanation method performance across 15 different evaluation metrics and three classification models in the chronic kidney disease dataset.** Each column represents an evaluation metric, and each row represents an explanation method. The scores for each metric are scaled between the minimum and maximum value, and methods are sorted by their average score. TreeExplainer outperforms previous approaches not only by having theoretical guarantees about consistency, but also by exhibiting improved performance across a large set of quantitative metrics that measure explanation quality (Methods). When these experiments were repeated for two synthetic datasets, TreeExplainer remained the top-performing method. Note that, as predicted, Saabas better approximates the Shapley values (and so becomes a better attribution method) as the number of trees increases (Methods). *Since MAPLE models the local gradient of a function, and not the impact of hiding a feature, it tends to perform poorly on these feature importance metrics [153].

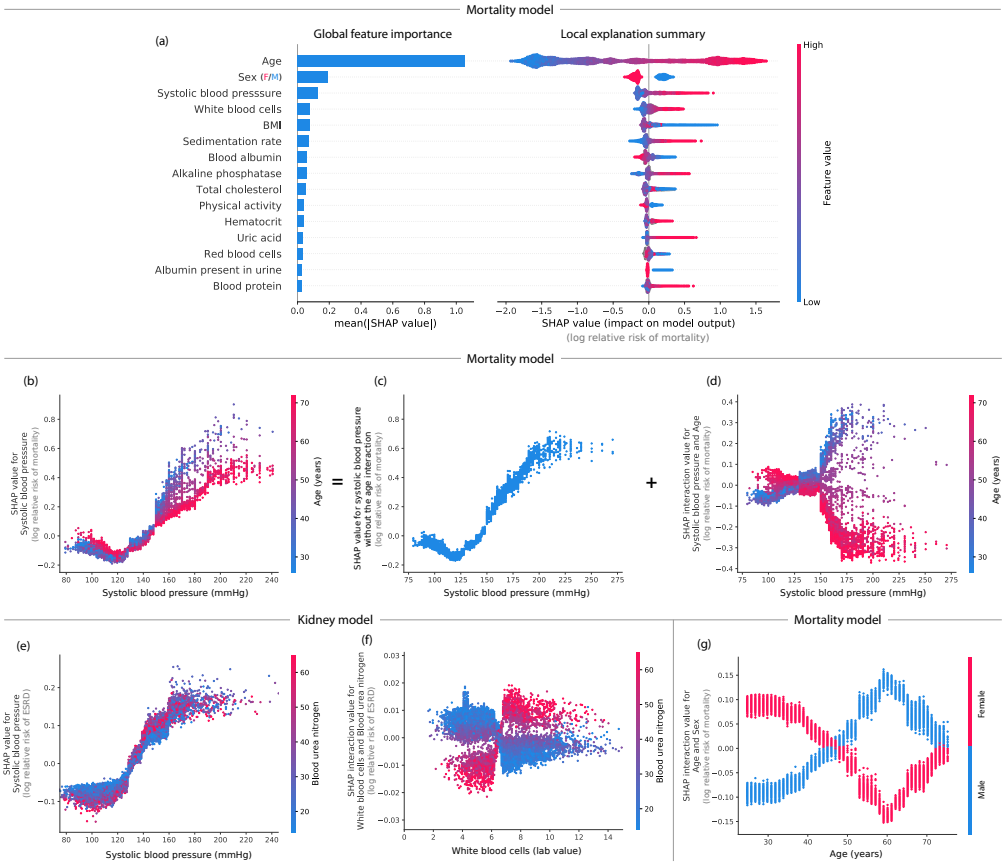


Figure 4.5: **By combining many local explanations, we can provide rich summaries of both an entire model and individual features.** Explanations are based on a gradient boosted decision tree model trained on the mortality dataset. (a) (left) bar-chart of the average SHAP value magnitude, and (right) a set of beeswarm plots, where each dot corresponds to an individual person in the study. The dot's position on the x-axis shows the impact that feature has on the model's prediction for that person. When multiple dots land at the same x position, they pile up to show density. (b) SHAP dependence plot of systolic blood pressure vs. its SHAP value in the mortality model. A clear interaction effect with age is visible, which increases the impact of early onset high blood pressure. (c) Using SHAP interaction values to remove the interaction effect of age from the model. (d) Plot of just the interaction effect of systolic blood pressure with age; shows how the effect of systolic blood pressure on mortality risk varies with age. Adding the y-values of C and D produces B. (e) A dependence plot of systolic blood pressure vs. its SHAP value in the kidney model; shows an increase in kidney disease risk at a systolic blood pressure of 125 (which parallels the increase in mortality risk). (f) Plot of the SHAP interaction value of 'white blood cells' with 'blood urea nitrogen'; shows that high white blood cell counts increase the negative risk conferred by high blood urea nitrogen. (g) Plot of the SHAP interaction value of sex vs. age in the mortality model; shows how the differential risk for men and women changes over a lifetime.

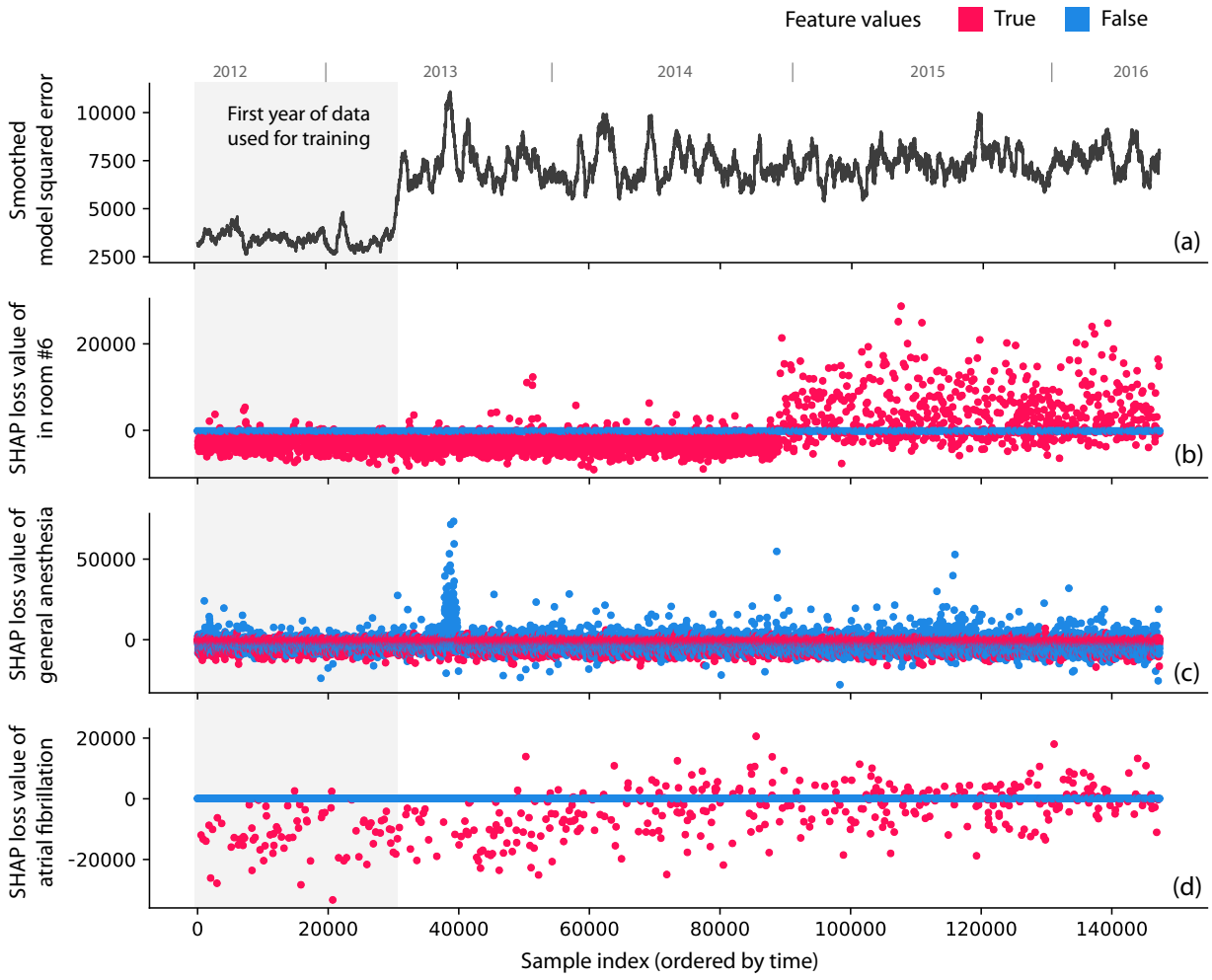


Figure 4.6: **Monitoring plots reveal problems that would otherwise be invisible in a retrospective hospital machine learning model deployment.** (a) The squared error of a hospital duration model averaged over the nearest 1,000 samples. The increase in error after training occurs because the test error is (as expected) higher than the training error. (b) The SHAP value of the model loss for the feature that indicates whether the procedure happens in room 6. A significant change occurs when we intentionally swap the labels of rooms 6 and 13, which is invisible in the overall model loss. (c) The SHAP value of the model loss for the general anesthesia feature; the spike one-third of the way into the data results from previously unrecognized transient data corruption at a hospital. (d) The SHAP value of the model loss for the atrial fibrillation feature. The plot's upward trend shows feature drift over time (P-value 5.4×10^{-19}).

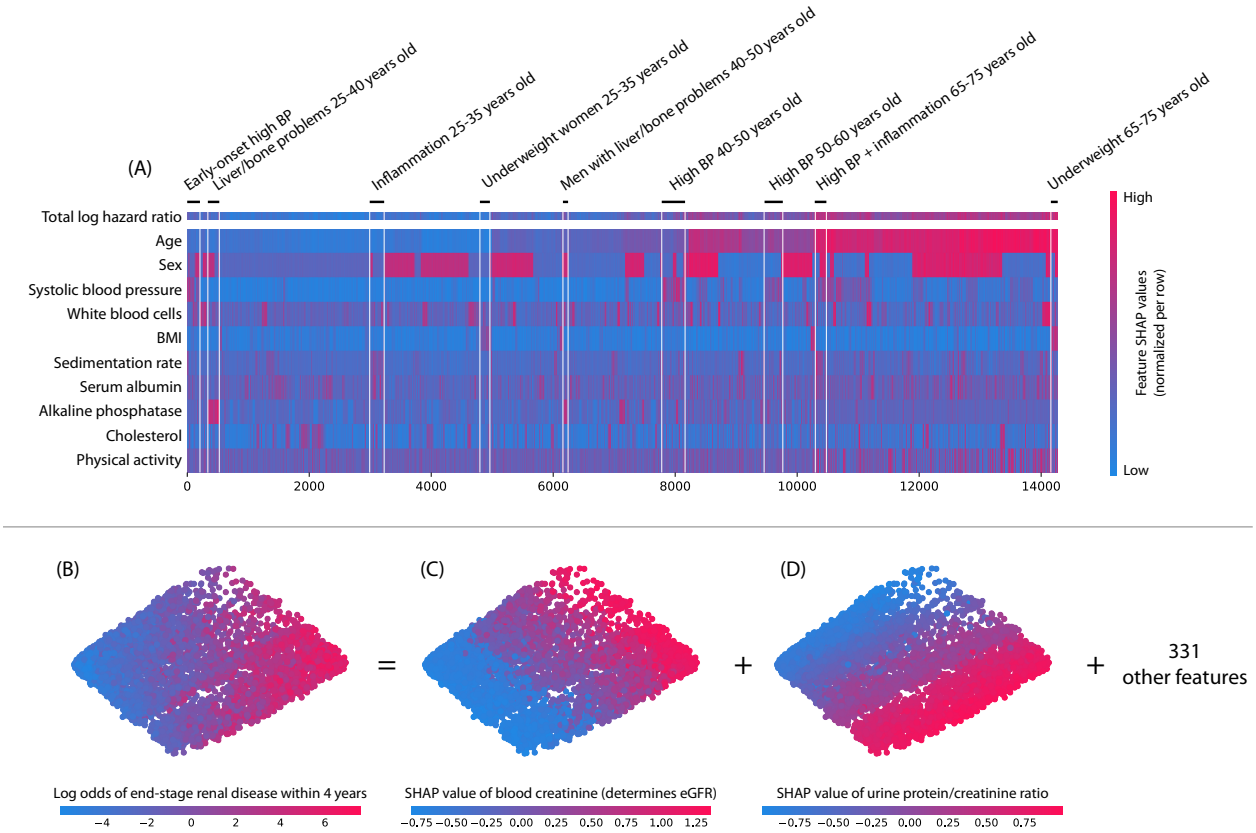


Figure 4.7: **Local explanation embeddings support both supervised clustering and interpretable dimensionality reduction.** (A) A clustering of mortality study individuals by their local explanation embedding. Columns are patients, and rows are features' normalized SHAP values. Sorting by a hierarchical clustering reveals population subgroups that have distinct mortality risk factors. (B-D) A local explanation embedding of kidney study visits projected onto two principal components. Local feature attribution values can be viewed as an embedding of the samples into a space where each dimension corresponds to a feature and all axes have the units of the model's output. The embedding colored by: (B) the predicted log odds of a participant developing end-stage renal disease within 4 years of that visit, (C) the SHAP value of blood creatinine, and (D) the SHAP value of the urine protein/creatinine ratio. Many other features also align with these top two principal components, and an equivalent unsupervised PCA embedding is far less interpretable.

Chapter 5

SELF-SUPERVISED LEARNING

5.1 Introduction

Globally, the number of surgical operations performed each year exceeds 300 million [205]. Although surgeries are crucial components of medical care, they have a higher prevalence of adverse events (i.e., patients harmed as a result of their medical treatment) relative to other medical specialties. In fact, several international studies have shown rates of adverse events ranging from 3% to 22% in surgical patients [143, 215, 93]. Fortunately, these studies also conclude that the majority of adverse events are preventable, indicating a tremendous opportunity for improvement by predictive models.

The accuracy of such models is largely dependent on the availability of training data. As of 2014, a large portion ($> 40\%$) of invasive, therapeutic surgeries take place in hospitals with either medium or small numbers of beds [181, 207]. These smaller institutions may lack either sufficient data or computational resources to train accurate models. Furthermore, patient privacy considerations mean that large public EHR datasets are unlikely, leaving many institutions with insufficient resources to train performant models on their own. In the face of this insufficiency, one natural way to make accurate predictions is *transfer learning*, which has already shown success in medical images as well as clinical text [190, 157, 124]. Particularly with the popularization of wearable sensors for health monitoring [125], transfer learning techniques that train models in one dataset and use them in another are arguably underexplored for physiological signals, which account for a significant portion of the hundreds of petabytes of currently available worldwide health data [162, 147]. One promising avenue of transfer learning research is *deep embedding models* which learn to extract generalizable features from images or time series data [36, 127] which improve over traditional

domain-specific hand engineered features.

Our approach, PHASE (PHysiologicAl Signal Embeddings), trains deep embedding models on physiological signals to better forecast and facilitate prevention of potentially millions of adverse surgical outcomes. Furthermore, these models not only improve predictive accuracy but can be transferred from an institution with plentiful computational resources to institutions with less. PHASE improves over previous approaches in two important ways:

- PHASE *improves predictive accuracy* by leveraging deep learning to embed physiological signals. Using long-short term memory networks (LSTMs), PHASE embeds physiological signals prior to forecasting adverse events with a downstream model. We investigate a number of self-supervised approaches (training with inputs and outputs derived from the signal data itself) [105] to effectively train embedding models. Our results show that gradient boosted tree (GBT) models trained with features extracted by self-supervised LSTMs improves accuracy over conventional approaches for forecasting surgical outcomes that rely on a single model (i.e., predicting adverse outcomes with an LSTM with raw features or a GBT with raw or hand engineered features).

- PHASE *shares models rather than data* to address data insufficiency and improves over alternative methods including GBTs trained with raw features, hand engineered features, and embeddings jointly learned by a single LSTM. Data insufficiency is especially important for surgical data because protecting patient privacy makes it difficult to share large amounts of medical data which exacerbates the lack of publicly available data [104]. By transferring performant models as has been done in medical images and clinical text [190, 157, 124], scientists can collaborate to improve accuracy of predictive models without exposing patient data.

In contrast to prior research on transfer learning for physiological signals that focus on a single medical center's electroencephalograms (EEGs) [58] or intensive care unit (ICU) stays [77], we evaluate transfer learning across three distinct medical center data sets (two from

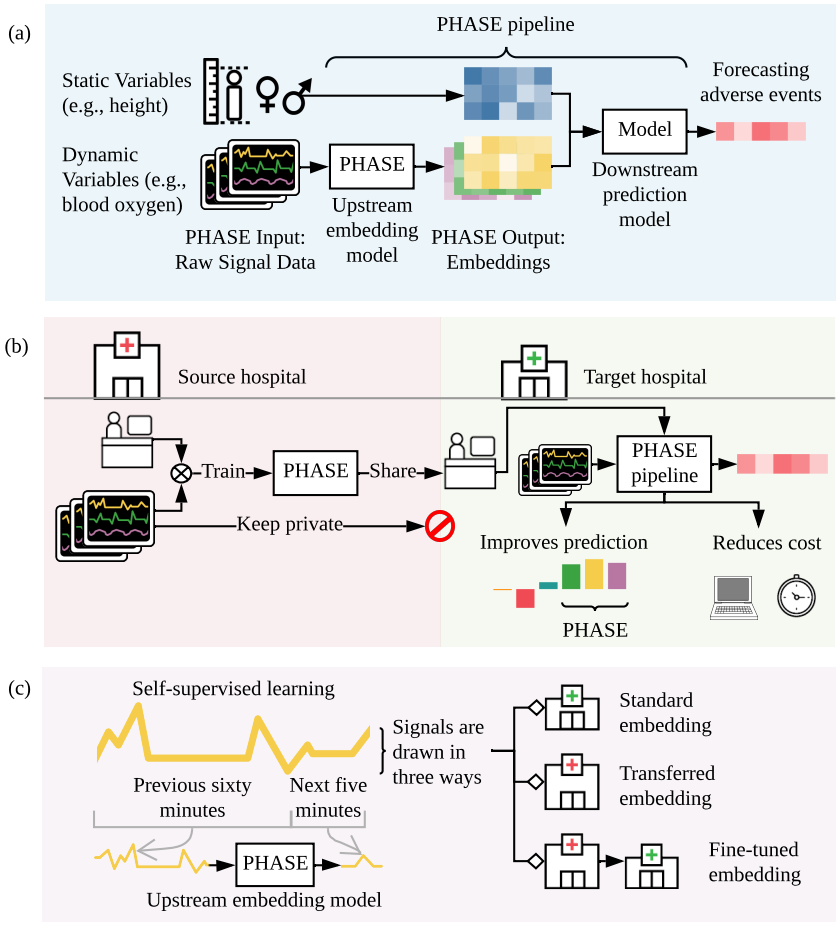


Figure 5.1: (a) PHASE learns models that embed (i.e., extract features from) physiological signals. We concatenate these embeddings with static data to predict adverse events. We describe the model extracting features as an *upstream embedding model* and the model making the final prediction as the *downstream prediction model*. (b) PHASE enables researchers at different hospitals to work together without sharing data. Researchers can perform transfer learning where upstream embedding models are trained on data drawn from a *source hospital* and used to embed signals and make a downstream prediction in data drawn from a *target hospital*. We show that this approach outperforms conventional deep learning and tree models trained with raw or hand engineered features. In addition, this approach reduces computational cost for users in target hospitals. (c) PHASE comprises LSTM embedding models trained per physiological signal that predict the future of the signal based on the past (self-supervised learning). We train self-supervised embedding models using data drawn in three distinct ways: (1) from the target hospital (standard embedding), (2) from a distinct source hospital (transferred embedding), and (3) from a distinct source hospital and then the target hospital (fine-tuned embedding) (More details in Section 5.2.2).

operating rooms and one from an ICU). Furthermore, we focus on evaluating self-supervised approaches (Figure 5.1) to train embedding models that we validate with feature attributions. To achieve this, we use data collected by the Anesthesia Information Management System (AIMS) from two medical centers as well as the Medical Information Mart for Intensive Care (MIMIC-III) dataset [91]. We utilize fifteen physiological signal variables and six static variable inputs (variables listed in Section 5.2.1) to forecast five possible outcomes: hypoxemia, hypocapnia, hypotension, hypertension, and phenylephrine administration. We show in a standard embedding setting, PHASE outperforms a number of conventional approaches across five outcomes of interest: hypoxemia, hypocapnia, hypotension, hypertension, and phenylephrine administration. Our results suggest that if the previous state of the art machine learning model (a gradient boosted tree model using hand engineered features [122]) captured 15% of hypoxemic events, PHASE captures approximately 19% of hypoxemic events based on a fixed recall. In our dataset we observe approximately 2.3 hypoxemic events per surgery, in the US alone our method could forecast roughly 1 million hypoxemic events that the previous state of the art model fails to capture (given that there are an annual 10 million surgeries in the US).

Furthermore, we show that PHASE improves performance in a transferred embedding setting where LSTM embedding models are trained in one dataset and used to extract features in a completely unseen dataset. Building upon this finding, we show that fine-tuning the LSTMs on unseen data leads to faster convergence and improved predictive performance compared to randomly initialized models across all outcomes. Finally, we validate our models by identifying important variables using state of the art local feature attribution methods [123]. We interpret our models to validate that the models uncover statistical patterns that agree with prior literature and demonstrate that models trained using PHASE are explainable. Importantly, explainability ensures that models are fair, trustworthy, and valuable to scientific understanding [52]. PHASE takes a step in the direction of allowing scientists to collaborate on EHR data which is typically accessible by only a single group (data silos [155]) by investigating approaches to train embedding models that generalize to unseen data.

5.2 Results

5.2.1 Five perioperative outcomes from three hospital datasets

We are interested in forecasting five outcomes associated with surgical morbidity. The first is hypoxemia (i.e., low blood oxygen level), an important cause of anesthesia-related morbidity, resulting in harmful effects on nearly every end organ [106, 56]. The next three outcomes are hypocapnia (i.e., low blood carbon dioxide), hypotension (i.e., low blood pressure), and hypertension (high blood pressure). Negative physiological effects associated with hypocapnia include reduced cerebral blood flow and reduced cardiac output [154]. Prolonged episodes of perioperative hypotension are associated with end-organ ischemia as well as assorted other adverse postoperative complications [116, 27, 89]. In addition, perioperative hypertension has been tied to increased risk of postoperative intracranial hemorrhage in craniotomies [15] and end organ dysfunction [200]. Finally, the last outcome of interest is the administration of phenylephrine, a medication frequently used to treat hypotension during anesthesia administration [97]. Predicting phenylephrine use lets us further evaluate PHASE because it represents a clinical decision rather than an aspect of patient physiology as in the prior four outcomes.

To evaluate our methodology with these outcomes, we utilize data from three different hospital datasets, summarized in Table 5.1 (Methods Section 5.4.1). In brief, we consider two operating room datasets from distinct medical centers which we denote as OR_0 and OR_1. We also use the publicly available intensive care unit MIMIC-III dataset which we refer to as ICU_M [91]. As inputs, we use fifteen physiological signal variables: *SAO2* - Blood oxygen saturation, *ETCO2* - End-tidal carbon dioxide, *NIBP[S/M/D]* - Non-invasive blood pressure (systolic, mean, diastolic), *FIO2* - Fraction of inspired oxygen, *ETSEV/ETSEVO* - End-tidal sevoflurane, *ECGRATE* - Heart rate from ECG, *PEAK* - Peak ventilator pressure, *PEEP* - Positive end-expiratory pressure, *PIP* - Peak inspiratory pressure, *RESPRATE* - Respiration rate, *TEMP1* - Body temperature in addition to six static variables: *Height*, *Weight*, *ASA Code*, *ASA Code Emergency*, *Gender*, and *Age*. All variables are consistently measured in

	Dataset	OR_0	OR_1	ICU_M
	Department	OR	OR	ICU
	Number of procedures/stays	29,035	28,136	1,669
Static variables	Gender (% female)	57%	38%	44%
	Age (yr) Mean	51.859	48.701	63.956
	Age (yr) Std.	16.748	18.419	17.708
	Weight (lb) Mean	185.273	181.608	176.662
	Weight (lb) Std.	54.042	54.194	55.448
	Height (in) Mean	66.913	67.502	66.967
	Height (in) Std.	8.268	8.607	6.181
	ASA Code Emergency	7.65%	15.31%	-
Adverse outcomes	Hypoxemia Base Rate	1.09%	2.19%	3.93%
	Hypocapnia Base Rate	9.76%	8.06%	-
	Hypotension Base Rate	7.44%	3.53%	-
	Hypertension Base Rate	1.70%	1.66%	-
	Phenylephrine Base Rate	7.23%	9.15%	-

Table 5.1: Training set statistics for different data sources. Each outcome has a different number of samples due to missing data.

the operating room datasets, but only *SAO2* is consistently measured in the ICU dataset.

Our metric of evaluation is the area under a precision recall curve, otherwise known as average precision (AP), which is more informative than the area under a receiver operating curve (ROC AUC) for binary predictions with low base rates [48], as in the outcomes we consider. In particular, we focus on the percent improvement over using the raw, unprocessed physiological signals as an evaluation metric, which is analogous to transfer loss: the difference between the transfer error and the in-domain baseline error [72]. We additionally report the absolute value of the AP (and ROC AUC for a subset of results).

5.2.2 Overview of the PHASE framework

PHASE is an approach to embed physiological signals. We consider an embedding framework using *upstream embedding models* U that are trained for each physiological signal in a source hospital data set H_s. We evaluate upstream embedding models with a downstream prediction

model D whose inputs are the embedded physiological signals concatenated to static variables and outputs are adverse surgical outcomes. D is trained in a target hospital data set H_t. We evaluate our models in three ways (Figure 5.1c): (1) standard embedding where the source hospital is the same as the target hospital $H_s = H_t$ (Figure 5.2b and Figure 5.2d), (2) transferred embedding where the source hospital is different to the target hospital $H_s \neq H_t$ (Figure 5.2c and Figure 5.2d), and (3) fine-tuned embedding where the upstream embedding model is first trained to convergence in a different source hospital $H_s \neq H_t$ and then used to initialize a model that is trained to convergence in the target hospital $H_s = H_t$ (Figure 5.3).

The modeling decision of *per-signal* upstream embedding was driven by several advantages: (1) we showed that per-signal embedding models produce embeddings that outperform downstream prediction models trained on the raw signals or hand-engineered signal features (described in Section 2.4) (2) we found that per-signal embedding models worked better than a single embedding model trained on all signals jointly in , and (3) we demonstrate that per-signal embedding models work even in a heterogeneous setting where the variables available in the target hospital are different to the variables available in the source hospital.

Here, we briefly describe the embeddings: *raw, ema, rand, auto, next, min*, and *hypo* in Figure 5.2a (more details in Methods Section 5.4.2). *Raw* and *ema* are not deep learning models. Instead, *raw* is the raw signal itself and *ema* are exponential moving averages and variance features from Lundberg et al. [122]. The remaining embeddings all use the final hidden layer of LSTMs trained in a source hospital H_s to embed the signals. The first embedding is *rand*, which uses an untrained LSTM with random weights. The second is an unsupervised approach called *auto*, which uses an LSTM trained to autoencode the input. The following two approaches (*next* and *min*) are self-supervised: the LSTM outputs are drawn from the same physiological signal variable as the input, but are taken from different parts of the signal. *Next* uses LSTMs trained to predict the next five minutes of a particular signal; *min* uses LSTMs trained to predict the minimum of the next five minutes of a particular signal. The final approach, *hypo*, is a traditional supervised approach to transfer learning where the embedding model has the same output as the downstream prediction

model (either hypoxemia, hypocapnia, or hypotension).

5.2.3 Comparing approaches to embed physiological signals

As a start, we first compare two popular machine learning models (GBTs and LSTMs) trained on the raw signal data (i.e., without embedding) concatenated to static patient data. In this section we will refer to results according to (1) the downstream model type and (2) the signal embedding type (for instance, GBT *raw* denotes a gradient boosted tree model trained with the raw minute by minute signal data). In Figure 5.2b, GBT *raw* performs comparably to LSTM *raw* for hypoxemia and better for hypocapnia and hypotension even though the LSTM should be more suitable to the time series signal data. Based on prior literature, we hypothesize that the GBT better captures patterns in the static patient data which is tabular [123], but the LSTMs better capture patterns in the time series data. In order to leverage the advantages of both model types, we propose PHASE which utilizes LSTMs to embed physiological signals and GBTs to perform the final prediction using the extracted features concatentated to static patient data (Figure 5.1a). In the following sections we primarily use GBTs as the downstream model and when we refer to our results solely by the signal data embedding they are assumed to use GBTs as the downstream model (for instance, *next* denotes a GBT model trained with *next* embedded data).

We first evaluate the PHASE methods that include two self-supervised embeddings (*next* and *min*) and a supervised embedding (*hypo*) in a standard embedding setting where the source dataset is the same as the target dataset (Figure 5.2b). We train GBT downstream models on the physiological signal embeddings concatenated to static patient features to see if the embeddings are more informative than the raw signals. *Rand* (which serves as a lower bound) transforms physiological signals in an uninformative manner and makes it harder to predict the outcomes of interest in comparison to the raw signals. Furthermore, *ema* and *auto* fail to consistently improve or impair performance relative to *raw* and thus are not viable features. In contrast, the PHASE methods (*next*, *min*, and *hypo*) consistently yield models that outperform the alternative approaches across all three outcomes (all p-values < 0.05).

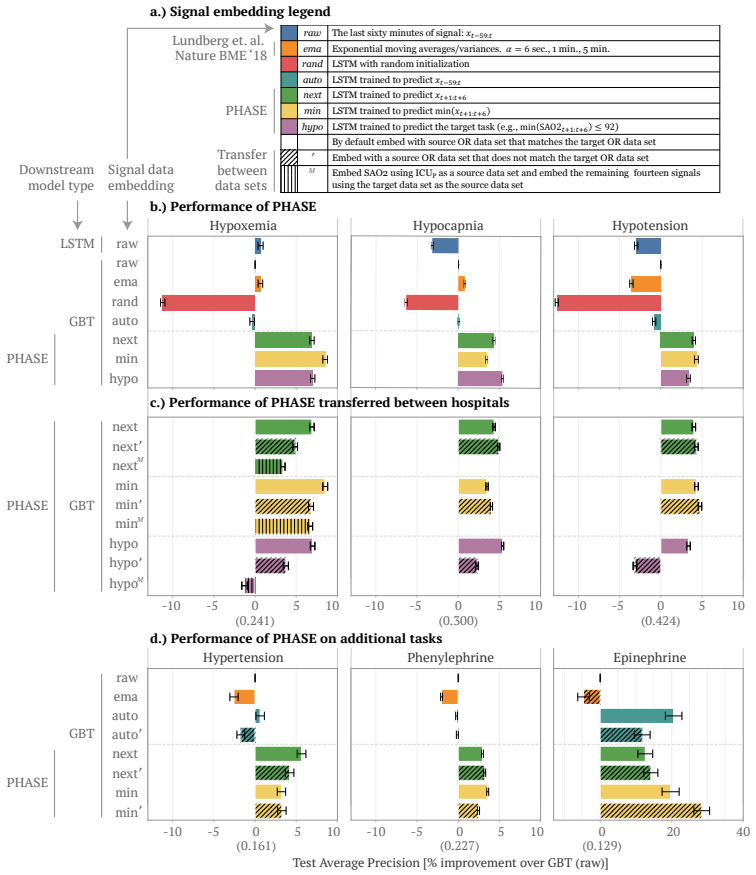


Figure 5.2: Comparing the performance of downstream models trained with different embeddings of physiological signals concatenated to static features. We report the average precision (% improvement over GBT model trained with *raw* signal data, 99% confidence intervals from bootstrapping the test set). We use OR_0 and OR_1 as target datasets and then aggregate across both by averaging the resultant means and standard errors of the % improvement. (a) The upstream embedding models we use to extract the physiological signal features where raw is the identify function, ema is an exponential moving average, and the rest are LSTMs trained in specific ways. (b) The performance of downstream prediction models for a variety of standard embedding approaches (when the source hospital is the same as the target hospital). We compare combinations of downstream models and embeddings for three adverse surgical outcomes (hypoxemia, hypocapnia, and hypotension). (c) The performance of transferred embedding (*next*', $next^M$, *min*', min^M, *hypo*', and $hypo^M$) vs. non-transferred (*next*, *min*, and *hypo*) models for the above three adverse outcomes. In the transferred approaches the source hospital is different to the target hospital. (d) Performance of approaches for standard and transferred embedding on additional outcomes: hypertension (high, rather than low, blood pressure) and phenylephrine (doctor action prediction). We do not evaluate *hypo* embeddings in this setting, because the outcomes are not "hypo" events.

In particular, *ema* is a gradient boosted tree model trained with hand engineered features (exponential moving average) shown to be on par with practicing anesthesiologists at forecasting hypoxemia (Lundberg et. al. Nature BME 2018 [122]). PHASE embeddings further improve over this approach suggesting that PHASE outperforms clinicians for forecasting hypoxemia by approximately 5% (Figure 5.2b).

In order to see how the choice of embedding model output affects downstream model performance we can take a closer look at *auto*, *next*, *min*, and *hypo*. Contrasting PHASE embeddings to *auto* suggests that *incorporating the future in the source task is crucial* (as in *next*, *min*, and *hypo*). However, while taking the minimum (*min*) and thresholding (*hypo*) make the upstream embedding model's outcome more similar to the downstream prediction model's outcome, *min* and *hypo* embeddings do not consistently improve downstream prediction performance compared to *next*.

The previously described results show that PHASE works when forecasting hypoxemia, hypocapnia, and hypotension; however these outcomes are all associated with low signals (hence the "hypo" prefix). In order to validate that PHASE performs well for "non-hypo" outcomes as well, we consider two additional outcomes: hypertension (i.e., high blood pressure) and phenylephrine administration (doctor action prediction) (Figure 5.2d). For hypertension we empirically demonstrate that *next* embeddings are better then *min* embeddings. This is to be expected because *min* focuses on the minimum of the future signal, whereas hypertension is defined as blood pressure being too high and it therefore addresses the maximum of the future signal. For phenylephrine, both the *next* and *min* models improve over standard approaches. One potential reason is that phenylephrine is typically administered in response to low blood pressure and thus *min* models are relevant to phenylephrine administration.

5.2.4 Evaluating upstream embedding models on unseen data

Previously we focused on a standard embedding setting in a single medical center; in this section, we examine the performance of PHASE when the upstream LSTM embedding models

are trained in one dataset but used to embed signals in an unseen dataset (i.e., *transferred* embedding setting). We analyze two distinct transfer learning settings where the source hospital differs to the target hospital (more details in Methods Section 5.4.5). We utilize a superscript notation ($'$ and ^{M}) to denote transfer learning. The apostrophe ($'$) denotes that we trained LSTMs in one operating room dataset and then fixed them to embed signal variables and evaluate performance with a downstream GBT model in the other. The superscript M (^{M}) denotes that we trained the LSTM for SAO2 in ICU_M and the other LSTMs in the target dataset[1].

Training the LSTM embedding models on a source dataset that differs from the target dataset and using a GBT downstream model ($'$ and ^{M} in Figure 5.2c and Figure 5.2d) generally outperforms conventional approaches: the LSTM trained on raw data and the GBT trained on raw or engineered features (LSTM *raw*, GBT *raw*, and *ema* in Figure 5.2b and Figure 5.2d). The *next* and *min* embeddings in the transferred embedding settings (*next*$'$, *min*$'$, *next*^{M}, *min*^{M}) outperform the conventional approaches for all possible outcomes (Figure 5.2c) including hypertension and phenylephrine (Figure 5.2d). However, for *hypo*, the supervised embedding, *hypo*$'$ improves over *raw* embeddings for hypoxemia and hypocapnia, but actually hurts performance for hypotension. Furthermore the *hypo*^{M} embedding also hurts performance for hypoxemia relative to using the *raw* embedding. This suggests that the choice of LSTM embedding model output is important and the supervised learning outcome (*hypo*$'$, *hypo*^{M}) does not generalize to unseen data as well as the self-supervised approaches (*next*$'$, *next*^{M}, *min*$'$, *min*^{M}).

Comparing the transferred embedding models ($'$ and ^{M} in Figure 5.2c and Figure 5.2d) to the standard embedding models (*next*, *min*, *hypo* in Figure 5.2c and Figure 5.2d) we see that the transferred embedding models generally perform comparably to the standard embedding

[1]MIMIC-III (ICU_m) has high rates of missingness for signals except for ECG (which is not directly present in the OR datasets) and SAO2. This means we were able to train an upstream LSTM only for SAO2 from ICU_m and we extracted features from the remaining signals using LSTMs trained in the target domain. This result is still meaningful, because it means we can use upstream embedding models trained in different domains synergistically.

models even though they are evaluated on previously unseen data. In particular, we see that the $next'$, min', $next^M$, and min^M embeddings perform comparably to their standard, non-transferred counterparts (*next* and *min*). It is worth noting that the transferred embeddings are equally performant for hypocapnia and hypotension; however, slightly reduce downstream performance for hypoxemia and hypertension, which may be due to differences in the hospital data sets (e.g., covariate shift). As before, we see that the $hypo'$ and $hypo^M$ embeddings perform substantially worse than their non-transferred counterpart *hypo*.

Although transferred PHASE embeddings perform slightly worse in the hypoxemia and hypertension prediction settings, one important advantage of transferring models is that end users in the target domain can use them at *no additional training cost*. Training all upstream LSTM embedding models for *next* constituted roughly 66 hours on an NVIDIA GeForce RTX 2080 Ti GPU. Clinicians who lack either computational resources or deep learning expertise to train their own models from scratch can instead use an off-the-shelf, fixed embedding model. Given that machine learning is usually not the primary concern of hospital staff, fixed embedding models are a straightforward way to improve the performance of models trained on physiological signal data at minimal cost to the end users.

There are two additional considerations for transfer learning: (1) In our results, we focus on evaluation using GBT downstream models. In order to show that the features we extract consistently boost performance and are robust to the choice of the downstream model we replicate our results for a multilayer perceptron (MLP) downstream model. (2) Per-signal LSTM embedding models outperform a single LSTM embedding model jointly trained with all signals. However, per-signal embedding models have an additional advantage: they work even when the variables available in the target hospital do not exactly match the ones in the source hospital (*feature heterogeneity*). Per-signal LSTM embedding models work in heterogeneous settings because end users can pick and choose models that correspond to the signals available at their institution. In comparison, a model trained on all possible variables would be unusable on a new hospital dataset with different variables. We show that in heterogeneous settings where the target hospital has fewer features than the source hospital,

GBTs trained with PHASE consistently outperform GBTs trained with the raw signals.

5.2.5 Fine-tuning upstream embedding models improves performance and reduces computational cost

In Section 5.2.4 we discussed that using PHASE embedding models in the transferred embedding setting are preferable to the standard embedding setting in terms of training cost; however, the standard embedding models still showed slightly better performance for hypoxemia and hypertension. Alternatively, we propose a fine-tuned embedding approach where we assume an end user in the target hospital has been provided a pre-trained embedding model trained in a distinct source hospital. Fine-tuning posits that deep models initialized using pre-trained models from a separate domain work better than randomly initialized models [211]. We train PHASE in a fine-tuning setting where upstream embedding models are trained in an OR target hospital initialized using the weights from the best model from the other OR hospital data set (detailed setup in Methods Section 5.4.5).

We find that PHASE in the fine-tuned embedding setting boosts performance over both standard embedding (Section 5.2.3) and transferred embedding (Section 5.2.4) in Figure 5.3b. We focus on *next* for the following experiment because it performed and generalized well across most outcomes in previous sections. In Figure 5.3, we evaluate the convergence and performance of fine-tuning LSTM embedding models. Figure 5.3a shows the convergence of fine-tuned models. The top two rows fix OR_0 as the target dataset. Dark green lines show the convergence of a randomly initialized LSTM and light green show the convergence of an LSTM initialized using weights from the best model in OR_1. The bottom two rows show the analogous plots with OR_1 as the target dataset. In Figure 5.3a we see that fine-tuning LSTMs rather than training them from scratch consistently leads to much faster convergence. In Figure 5.3b, we see that LSTMs obtained from fine-tuning ($next^{\text{ft}}$) consistently outperform those trained in a single dataset: standard embeddings (*next*) and transferred embeddings ($next'$). These results indicate that end users can fine-tune PHASE LSTMs to boost performance at lower computational cost in comparison to training models from scratch. Although

fine-tuning is more computationally costly than a pre-trained model (transferred embedding), the performance gains from fine-tuning are more consistent.

5.2.6 Validating models with local feature attributions

We summarize key variables used by downstream GBT models using summary plots (Figure 5.4). In these plots, each point represents a feature's importance for a single sample, with the x-axis showing the feature's impact on the model's output and the colors indicates the feature's value (attribution method details in Methods Section 5.4.5). We focus on explaining GBT models trained on PHASE *next* embeddings in terms of each variable because *next* embeddings were performant across most of the outcomes we considered. The colors are the sum of all features associated with a single signal variable (200 extracted features) which are not naturally interpretable because the embedding values can be arbitrarily positive or negative based on the embedding models.

Standard approaches to train embedding models would use all signal variables as inputs to a single model. These approaches are harder to interpret, because each embedding dimension may be dependent on multiple signals simultaneously. Having per-signal embedding models as in PHASE allows us to clearly interpret each embedding as being dependent on a single physiological signal variable.

We validate important variables against prior literature for models trained on *next* embeddings for all five outcomes (Figure 5.4). For hypoxemia, the important variables includes variables logically connected to blood oxygen: *SAO2*, *ETCO2*, and *FIO2* are all associated with the respiratory system, while and *PIP* is tied to mechanical ventilation which is naturally linked to blood oxygen [98, 54]. For hypocapnia *ETCO2* is logically the most important feature. Furthermore, using *FIO2*, *RESPRATE*, *PIP*, and TV to forecast hypocapnia makes sense because these variables all relate to either ventilation or respiration. As one would expect, for hypotension and hypertension, key variables are generally the three non-invasive blood pressure measurements: *NIBPM*, *NIBPD*, *NIBPS*. Furthermore, a number of studies validate the importance of *ECGRATE* (heart rate measured from ECG signals) to forecast-

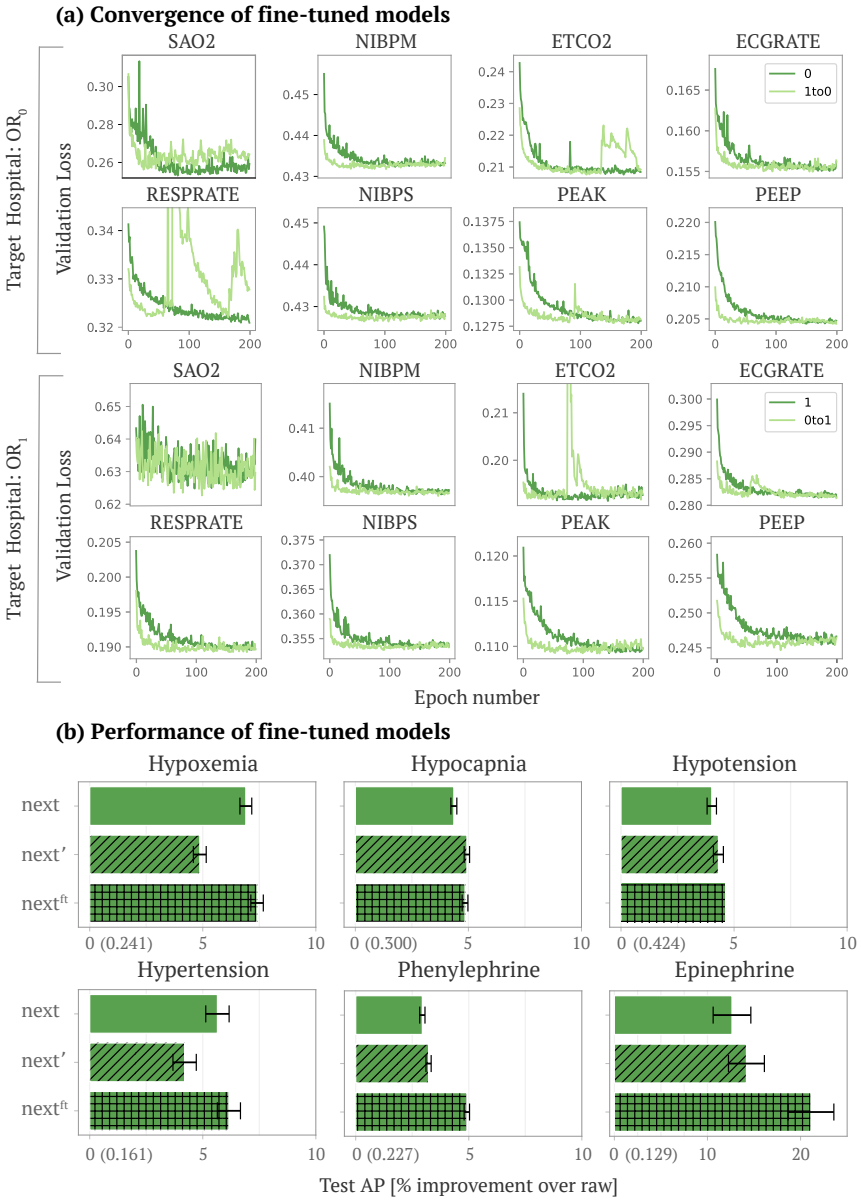


Figure 5.3: (a) The convergence of fine-tuned models. The top eight plots fix OR_0 as the target dataset (we plot eight out of the total fifteen signals). Dark green lines show the convergence of a randomly initialized LSTM trained in OR_0 and light green show the convergence of an LSTM trained in OR_0 initialized using weights from the best model in OR_1 (fine-tuning). The bottom two rows show the analogous plots with OR_1 as the target dataset. (b) The performance of GBT models trained on embeddings from standard embedding models (*next*), transferred embedding models (*next'*), and fine-tuned embedding models ($next^{ft}$) (best models from light green in (a)).

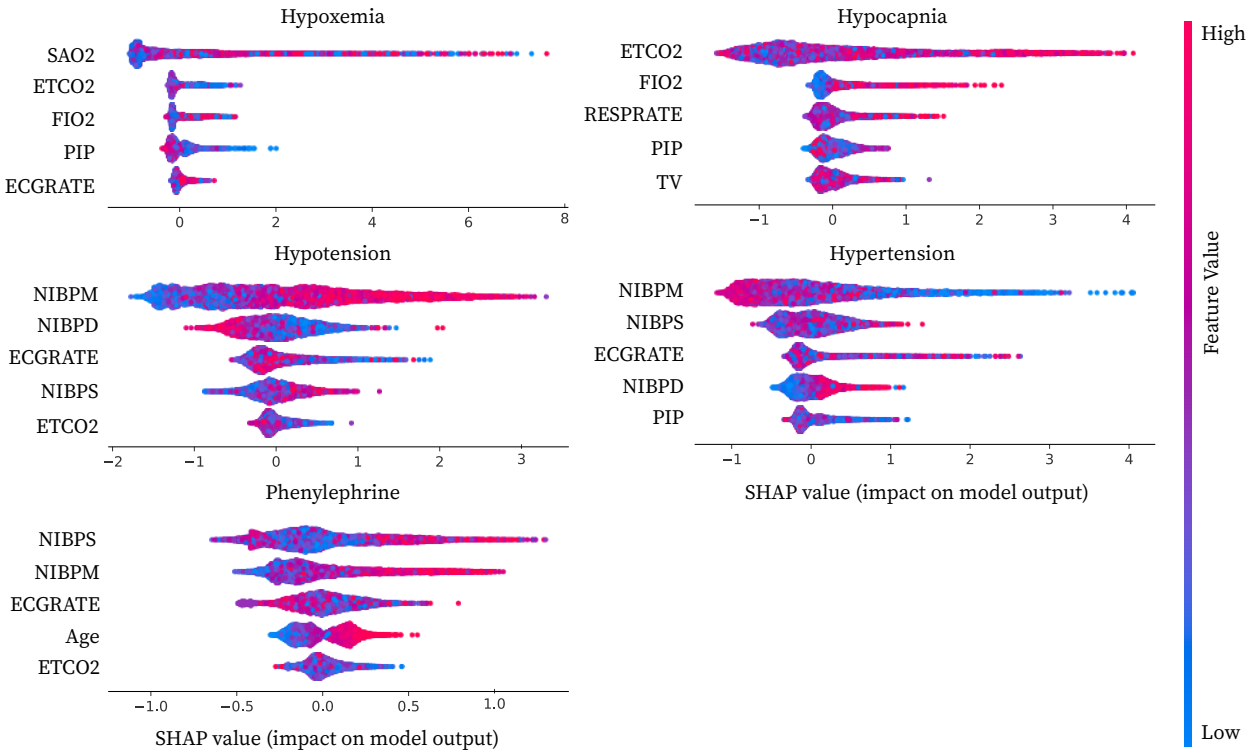


Figure 5.4: Local feature attribution summary plots for the top five most important variables from GBT models trained with *next* embeddings in the target dataset OR_0. In order to obtain attributions for each variable we explain each GBT using Interventional Tree Explainer. This gives us attributions for *next* embeddings for the fifteen physiological signal variables (200 dimensional embeddings for each) and six static variables. We sum over embedding attributions to obtain the importance of a particular physiological signal variable. On the x-axis we report this aggregated attribution value that indicates the variable's cumulative impact on the model output. The colors of the points are either the feature's value for static variables or the sum over all *next* embeddings for a given physiological signal variable.

ing hypotension and hypertension [149, 139]. Finally, phenelyphrine is typically administered during surgery in response to hypotension, thus validating the importance of *NIBPS*, *NIBPM*, and *ECGRATE*. Similarly, age being more important to forecast phenelyphrine use may be tied to its predictive relationship to hypotension as well as anesthesiologists' heightened vigilance to hypotension in the higher-risk older population [119].

5.3 Discussion

This study explored machine learning techniques for forecasting adverse surgical outcomes. Given the rates of adverse events in the operating room [143, 215, 93], computational forecasting that provides advanced warning may be of widespread utility to medical practitioners. This is especially the case given that the outcomes we considered (hypoxemia, hypocapnia, hypotension, and hypertension) are all tied to a number of harmful physiological effects.

This work also shows physiological signal embeddings are effective in several settings. We demonstrate that standard embedding using LSTMs improves the performance of downstream models (GBT and MLP), which implies that pipelines utilizing deep networks to embed physiological signals are effective for electronic healthcare record data. Next, we show that PHASE embedding models work almost equally well in a transferred embedding setting as in a standard embedding setting, and, in fact, work better than randomly initialized models if fine-tuned. This implies that sharing pre-trained networks can improve downstream models in terms of computational needs and predictive performance.

PHASE uses independently trained LSTMs for each signal variable. Surprisingly, we demonstrate that our per-signal approach outperforms a jointly trained embedding model LSTM. Furthermore, having each LSTM associated with a single physiological signal actually proves to be an advantage. Hospitals often collect different sets of physiological signal variables; to address this heterogeneity, target hospitals with different but overlapping variables to a source hospital can use the embedding models for the variables which they both have.

One limitation of PHASE is that although sharing models reveals less information than sharing data, it is possible to use model inversion attacks on the PHASE embedding models [64] to find physiological signals similar to the training data. Although we attempted to use differentially private versions of stochastic gradient descent [6] to train our embedding models, the randomness inserted in the training process made it difficult to train effective models. We leave investigation and development of effective privacy preserving techniques

to train such models to future work. Another limitation of our data is that the embedding models only apply to physiological signals sampled once per minute. We leave exploration of adapting models to accommodate multiple sampling frequencies to future work as well. Finally, it should be said that there is complementary work discussing deep learning for electrocardiograms [168, 129] and electroencephalograms [144]. We focus primarily on minute by minute physiological signals collected within an operating room setting. As such, although we do have an ECGRATE variable, we do not directly use the electrocardiogram signals.

Our work takes an important step forward in applying machine learning to the domain of physiological signals. Our approach differs from previous studies, which did not explore physiological signal transfer learning across multiple hospitals or analyze self-supervised approaches for training deep models.

Drawing on parallels from computer vision (CV) and natural language processing (NLP), both exemplars of transfer learning, physiological signals are well suited to neural network embeddings (i.e., transformations of original inputs into a space better suited to make predictions). In particular, CV and NLP share two notable traits with physiological signals. The first is *consistency*. The CV domain has consistent features: edges, colors, and other visual attributes [156, 173]; the NLP domain uses a particular language with semantic relationships consistent across bodies of text [38]. For sequential signals, we saw that physiological patterns are consistent, because PHASE generalized across hospitals in a transferred embedding setting. The second attribute is *complexity*. Each of these domains is sufficiently complex to make learning embeddings non-trivial. These factors suggest that individual research scientists must make redundant efforts to learn embeddings that may ultimately be very similar. To avoid this problem, NLP and CV have made significant progress on standardizing and evaluating their embeddings; in the health domain, physiological signals are a natural next step. More significantly, the use of physiological signals is constrained by patient privacy; this makes it difficult to share *data* between hospitals. However, sharing *models* between hospitals does not directly expose patient information. Sharing models in this way could allow machine learning for physiological signals to see similarly large advances as in computer

vision and natural language.

5.4 Methods

5.4.1 Datasets

The operating room (OR) datasets were collected via the Anesthesia Information Management System (AIMS), which includes static information as well as real-time measurements of physiological signals sampled minute by minute. OR_0 was drawn from an academic medical center and OR_1 from a trauma center. Two marked differences between the patient distributions of OR_0 and OR_1 are the gender ratio (57% females in the academic medical center versus 38% in the trauma center) and the proportion of ASA codes that are classified as emergencies (7.65% emergencies versus 15.31%). ICU_M is a sub-sampled version drawn from PhysioNet's publicly available MIMIC dataset, which contains data obtained from an intensive care unit (ICU) in Boston, Massachusetts [91]. Although ICU_M data contains several physiological signals sampled at a high frequency, we solely used a minute-by-minute *SAO2* signal for our experiments because other physiological signals had a substantial amount of missingness. Furthermore, ICU_M contained neonatal data that we filtered out. For all three datasets, any remaining missing values in the signal features were imputed by the mean, and each feature was standardized to have unit mean and variance for training neural networks. Additional details about the distributions of patients in all three datasets are shown in Table 5.1.

5.4.2 Set-up

For our datasets, we considered a distribution of hospital stays $\mathcal{P}$. Since we wanted to forecast an adverse event in time, we defined samples by first drawing a hospital stay $P \sim \mathcal{P}$ and then drawing a time point $t \sim (1, \cdots, len(P))$. For the rest of this set-up, we assume we are operating with samples i defined by t, P.

Variables

Many variables are associated with each hospital stay. We distinguished between static variables (that are constant throughout the course of a patient's stay and are solely determined by P) and dynamic variables (that change over time and are determined by P and t). We partition each sample i's (i is implicitly determined by P and t) variables into two distinct sets:

$$X^i = (\underbrace{X^i_{s_1}, \cdots, X^i_{s_6}}_{\text{Static variables}}, \underbrace{X^i_{d_1}, \cdots, X^i_{d_{15}}}_{\text{Dynamic variables}}) \tag{5.1}$$

The six static variables $(X^i_{s_1}, \cdots, X^i_{s_6})$ that do not change over the course of a surgery are: *Height*, *Weight*, *ASA Code*, *ASA Code Emergency*, *Gender*, and *Age*.

Furthermore, we utilized fifteen physiological signals for our dynamic variables $(X^i_{d_1}, \cdots, X^i_{d_{15}})$:

- *SAO2* - Blood oxygen saturation
- *ETCO2* - End-tidal carbon dioxide
- *NIBP[S/M/D]* - Non-invasive blood pressure (systolic, mean, diastolic)
- *FIO2* - Fraction of inspired oxygen
- *ETSEV/ETSEVO* - End-tidal sevoflurane
- *ECGRATE* - Heart rate from ECG
- *PEAK* - Peak ventilator pressure
- *PEEP* - Positive end-expiratory pressure
- *PIP* - Peak inspiratory pressure
- *RESPRATE* - Respiration rate
- *TEMP1* - Body temperature

- *PHENYL* - Whether phenylephrine was administered. We only use this as an output variable and not as an input.

To index the dynamic variables we used the following notation to denote minutes a to b (where $b > a$) of a particular signal:

$$X^i_{d_j}[a:b] \in \mathbb{R}^{b-a} \tag{5.2}$$

Outcomes

We focused on binary outcomes (i.e., downstream prediction tasks):

$$y^i \in \{0, 1\} \tag{5.3}$$

Our adverse events define the outcome as a function ($g(\cdot), e.g., g(\cdot) = min(\cdot) < C$) of the next five minutes of a physiological signal ($X^i_{d_j}$):

$$y^i = g(X^i_{d_j}[t+1:t+5]) \tag{5.4}$$

Specifically, we focused on health forecasting tasks; forecasting tasks facilitate preventive healthcare by helping healthcare providers mitigate risk preemptively [180]. In particular, we considered the following five tasks (which all focus on the next five minutes of surgery):

- *Hypoxemia:* was blood oxygen less than 93?

$$\min(X^i_{SAO2}[t+1:t+5]) < 93 \tag{5.5}$$

- *Hypocapnia:* was end tidal carbon dioxide less than 35?

$$\min(X^i_{ETCO2}[t+1:t+5]) < 35 \tag{5.6}$$

- *Hypotension:* was mean blood pressure less than 60?

$$\min(X^i_{NIBPM}[t+1:t+5]) < 60 \tag{5.7}$$

- *Hypertension:* was mean blood pressure higher than 110?

$$\min(X^i_{NIBPM}[t+1:t+5]) > 110 \tag{5.8}$$

- *Phenylephrine:* was phenylephrine administered?

$$\max(X^i_{PHENYL}[t+1:t+5]) = 1 \tag{5.9}$$

5.4.3 Embeddings (i.e. features)

We define variables (e.g., height, blood oxygen, etc.) separately from embeddings (e.g., height, minute 20 of blood oxygen, etc.) which the downstream prediction models are trained on. Notationally, we denote embeddings as lower case:

$$x^i = (x^i_{s_1}, \cdots, x^i_{s_6}, x^i_{d_1}, \cdots, x^i_{d_{15}}).$$

We embed the dynamic variables, with a function $U_{d_k;E}$ of the past sixty minutes of the physiological signal variable:

$$x^i_{d_k} = U_{d_k;E}(X^i_{d_k}[t-59:t]), \forall k \in 1, \cdots, 15, E \in \{raw, ema, rand, auto, next, min, hypo\}.$$

We use the static variables as is: $x^i_{s_k} = X^i_{s_k}, \forall k \in 1, \cdots, 6$. For GBT downstream models we do not transform the static variables; however, for the LSTM downstream models we do normalize them. Unlike dynamic variables, extracting features from the static variables does not significantly improve performance of downstream models.

5.4.4 Downstream prediction model

The downstream prediction models D are used to evaluate different types of embeddings. They are trained on the embedded samples x^i drawn from a target hospital H_t. D minimizes binary cross entropy loss to forecast adverse outcomes y^i defined as a function of the future five minutes of a physiological signal (for example hypoxemia would be $\min(X^i_{d_{SAO2}}[t+1 : t+5]) < 93$, where $X^i_{d_{SAO2}}[t+1 : t+5]$ denotes the future five minutes of the blood oxygen variable for sample i).

5.4.5 Dynamic embedding

For dynamic variables, we made two important decisions. The first was how much of the signal to use. To make fair comparisons, we gave all models access only to the 60 minutes of the signal prior to the outcome (which starts at $t+1$):

$$X^i_{d_j}[t-59 : t] \tag{5.10}$$

The second important decision was how to embed a signal ($X^i_{d_j}$). Two natural embeddings are: (1) to use the sixty minutes as is (*raw*):

$$x^i_{d_j} = X^i_{d_j}[t-59 : t] \in \mathbb{R}^{60} \tag{5.11}$$

Where $U_{d_j;raw}$ is the identify function.

and (2) to use exponential moving averages and variances as the embedding function $U_{d_j;ema}$ (*ema*) [122]:

$$x^i_{d_j} = (EMA(X^i_{d_j}[t-59 : t], \alpha = 0.1), EMA(X^i_{d_j}[t-59 : t], \alpha = 1), \tag{5.12}$$

$$EMA(X^i_{d_j}[t-59 : t], \alpha = 5), EMV(X^i_{d_j}[t-59 : t], \alpha = 5)) \in \mathbb{R}^4 \tag{5.13}$$

where the exponential moving average is defined as:

$$EMA_\tau = \alpha \times X^i_{d_j}[\tau] + (1-\alpha) \times EMA_{\tau-1}, \forall \tau > t-59 \tag{5.14}$$

$$EMA_{t-59} = X^i_{d_j}[t-59] \tag{5.15}$$

$$EMA(X^i_{d_j}[t-59:t], \alpha) = EMA_t \tag{5.16}$$

and the exponential moving variance is defined as:

$$\delta_\tau = X^i_{d_j}[\tau] - EMA_{\tau-1} \tag{5.17}$$

$$EMA_\tau = EMA_{\tau-1} + \alpha \times \delta_\tau \tag{5.18}$$

$$EMV_\tau = (1-\alpha) \times (EMV_{\tau-1} + \alpha \times \delta^2_\tau) \tag{5.19}$$

$$EMV(X^i_{d_j}[t-59:t], \alpha = 5) = EMV_t \tag{5.20}$$

LSTM embedding

To better extract features from (embed) each physiological signal variable ($X^i_{d_j}$), we utilized per-signal neural networks (LSTMs) trained in a source hospital H_s. The LSTMs $L^{H_s}_{d_j;E}$ are trained for each physiological signal to minimize a loss function (dependent on the embedding type E) with the past sixty minutes of signal d_k as the input:

$$\mathcal{L}_E(L^{H_s}_{d_k;E}(X^i_{d_k}[t-59:t]), y^i_E)$$

E	Domain	Range (Upstream Task)	$\mathcal{L}_E$
rand	$X^i_{d_j}[t-59:t] \in \mathbb{R}^{60}$	$\emptyset$	$\emptyset$
auto	$X^i_{d_j}[t-59:t] \in \mathbb{R}^{60}$	$X^i_{d_j}[t-59:t] \in \mathbb{R}^{60}$	MSE
next	$X^i_{d_j}[t-59:t] \in \mathbb{R}^{60}$	$X^i_{d_j}[t+1:t+5] \in \mathbb{R}^5$	MSE
min	$X^i_{d_j}[t-59:t] \in \mathbb{R}^{60}$	$\min(X^i_{d_j}[t+1:t+5]) \in \mathbb{R}^1$	MSE
hypo	$X^i_{d_j}[t-59:t] \in \mathbb{R}^{60}$	$y^i \in \{0,1\}$	BCE

Table 5.2: Inputs and outputs for our per-signal upstream LSTMs.

Table 5.2 describes the different tasks we used to train LSTMs upstream embedding models including the three self-supervised labels (*next*, *min*, *hypo*) we proposed in PHASE. More specifically, $U_{d_j;E} = h \circ L^{H_s}_{d_j;E}$, where the composition $h \circ L$ signifies removing the output layer of L to obtain a function that maps the past sixty minutes of d_k to the activations of the final hidden layer in L. For the *rand* embedding the models $L_{d_k;rand}$ are LSTM models with random weights. There is no source hospital, because the models are not trained. Then, *auto*, *next*, and *min* embeddings set $\mathcal{L}_E$ to mean squared error. However, the outcomes differ for each: $y^i_{auto} = X^i_{d_k}[t-59:t], y^i_{next} = X^i_{d_k}[t+1:t+5], y^i_{mind} = \min(X^i_{d_k}[t+1:t+5])$ (note that these outcomes are self-supervised). Finally, *hypo* embeddings set $\mathcal{L}_E$ to binary cross entropy loss and the outcome is set to be the same as the downstream task y^i. Since several of our downstream outcomes were tied to too-low ("hypo") signals, the approaches in Table 5.2 were ordered by distance to the downstream task.

We used the following notation to denote an LSTM trained to convergence using $X^i_{d_j}$ drawn from the source hospital dataset H_s using inputs and outputs specified by the task in Table 5.2:

$$L^{H_s}_{d_j;task} \tag{5.21}$$

As an example, $L^{\text{OR}_0}_{d_j;next}$ indicates that the LSTM was trained for signal $X^i_{d_j}$ with inputs $X^i_{d_j}[t-59:t]$ and outputs $X^i_{d_j}[t+1:t+5]$ on data drawn from OR_0.

To describe the features associated with the neural network embedding approaches, we removed the output layer of the network and embedded each signal using the final hidden layer of the network. We denote this as:

$$x^i_{d_j} \equiv h \circ L^{H_s}_{d_j;next}(X^i_{d_j}[t-59:t]) \in \mathbb{R}^{200} \tag{5.22}$$

where h removes the output layer of network L and 200 is the number of hidden nodes in L.

As an example, if our target dataset was OR_0, then our physiological signal features for

next would be:

$$x^i_{d_j} \equiv h \circ L^{\mathrm{OR}_0}_{d_j;next}(X^i_{d_j}[t-59:t]) \in \mathbb{R}^{200} \tag{5.23}$$

Transferred embedding

To evaluate transfer learning, we denoted a target hospital dataset H_t (the domain in which we trained the downstream prediction model on embedded variables) and a source hospital dataset H_s (the domain in which we trained our upstream embedding models). In the transference experiments (denoted used superscripts next to the embedding type E: $task'$ and $task^M$) we train our upstream embedding models in a source hospital that is different to the target hospital ($H_s \neq H_t$).

By default, without the superscript, the source domain matched the target domain ($H_s = H_t$). With an apostrophe, the source domain was the remaining operating room dataset ($H_s = \mathrm{OR}_0$ if $H_t = \mathrm{OR}_1$ or $H_s = \mathrm{OR}_1$ if $H_t = \mathrm{OR}_0$). As an example, if our target dataset was OR_0, then our physiological signal features for $next'$ would be:

$$x^i_{d_j} \equiv h \circ L^{\mathrm{OR}_1}_{d_j;next}(X^i_{d_j}[t-59:t]) \in \mathbb{R}^{200} \tag{5.24}$$

Finally, for $task^M$, the source domain for the LSTM embedding model for SAO2 was ICU_M ($H_s = \mathrm{ICU}_\mathrm{M}$), and the remaining models were trained in a source domain that matched the target domain ($H_s = H_t$). As an example, if our target dataset was OR_0, then our physiological signal features for $next'$ would be:

$$x^i_{d_j} \equiv h \circ L^{\mathrm{ICU}_\mathrm{M}}_{d_j;next}(X^i_{d_j}[t-59:t]) \in \mathbb{R}^{200} \text{ for SAO2} \tag{5.25}$$

$$x^i_{d_j} \equiv h \circ L^{\mathrm{OR}_0}_{d_j;next}(X^i_{d_j}[t-59:t]) \in \mathbb{R}^{200} \text{ for all other signals} \tag{5.26}$$

Fine-tuned embedding

The fine-tuning approach (denoted as next^{ft}) considers fine tuning models between operating room datasets. If we assume a fixed target dataset $H_t = \text{OR}_0$. Then, as before, we denote an LSTM trained to convergence on data from OR_1 to be:

$$L^{\text{OR}_1}_{d_j;next} \tag{5.27}$$

For fine-tuning, we used the LSTM trained on samples drawn from OR_1 (which crucially was not the same as the target dataset) to initialize an LSTM which we then trained until convergence on samples drawn from OR_0. Notationally, we describe this as:

$$L^{\text{OR}_1 \to \text{OR}_0}_{d_j;next} \tag{5.28}$$

The features for dynamic variables under the fine-tuning approach for $H_t = \text{OR}_0$ were:

$$x^i_{d_j} \equiv h \circ L^{\text{OR}_1 \to \text{OR}_0}_{d_j;next}(X^i_{d_j}[t-59:t]) \in \mathbb{R}^{200} \tag{5.29}$$

Jointly Trained Upstream Model

The jointly trained upstream model (denoted as $next_{\text{m}}$) involved training an LSTM for several signals simultaneously. To do so, we optimized an LSTM for forecasting the next five minutes of all our physiological signals, which we denote as:

$$L^{H_s}_{d_1,\cdots,d_{15};next} \tag{5.30}$$

Then, the features for dynamic variables under the jointly trained multi-signal model were:

$$x^i_{d_1},\cdots,x^i_{d_{15}} = h \circ L^{H_s}_{d_1,\cdots,d_{15};next}(X^i_{d_1}[t-59:t],\cdots,X^i_{d_{15}}[t-59:t]) \tag{5.31}$$

Local Feature Attributions

To obtain explanations, we utilized Interventional Tree Explainer, which provides exact Shapley values with an interventional conditional (marginal) expectation set function (feature attributions with game-theoretic properties) for complex tree-based models [123, 121]. The Shapley values serve as local feature attributions $\phi(f, x^i)$ that indicate how much each feature in x^i contributed to a single downstream prediction $D(x^i)$. Positive attribution means that the feature generally increases the output of the model (risk of adverse events) and negative attribution means that the feature generally decreases the output. Shapley values have been used to explain models in a wide variety of applications including biology [100], medicine [150], finance [194], and more.

Chapter 6

EXPLAINING DEEP MODELS/A SERIES OF MODELS

6.1 Introduction

With the widespread adoption of machine learning (ML), *series of models* (i.e., where the outputs of predictive models are used as inputs to separate predictive models) are increasingly common. Examples include: *stacked generalization*, a widely used technique [204, 79, 18, 53, 2] to improve generalization performance by ensembling the predictions of many models (called base-learners) using another model (called a meta-learner) [209], *neural network feature extraction*, where models are trained on features extracted using neural networks [76, 36], typically for structured data [210, 115, 85], and *consumer scores*, where predictive models that describe a specific behavior (e.g., credit scores [51]) are used as inputs to downstream predictive models. For example, a bank may use a model to predict customers' loan eligibility on the basis of their bank statements and their credit score, which itself is often a predictive model [62].

Explaining a series of models is crucial for debugging and building trust, even more so because a series of models is inherently harder to explain compared to a single model. One popular paradigm for explaining models are *local feature attributions*, which explain why a model makes a prediction for a single sample (known as the *explicand* [188]). Existing *model-agnostic* local feature attribution methods (e.g., IME [183], LIME [160], KernelSHAP [121]) work regardless of the specific model being explained. They can explain a series of models, but suffer from two distinct shortcomings: (1) their sampling-based estimates of feature importance are inherently variable, and (2) they have high computational cost which may not be tractable for large pipelines. Alternatively, *model-specific* local feature attribution methods (i.e. attribution methods that work for specific types of models) are often much

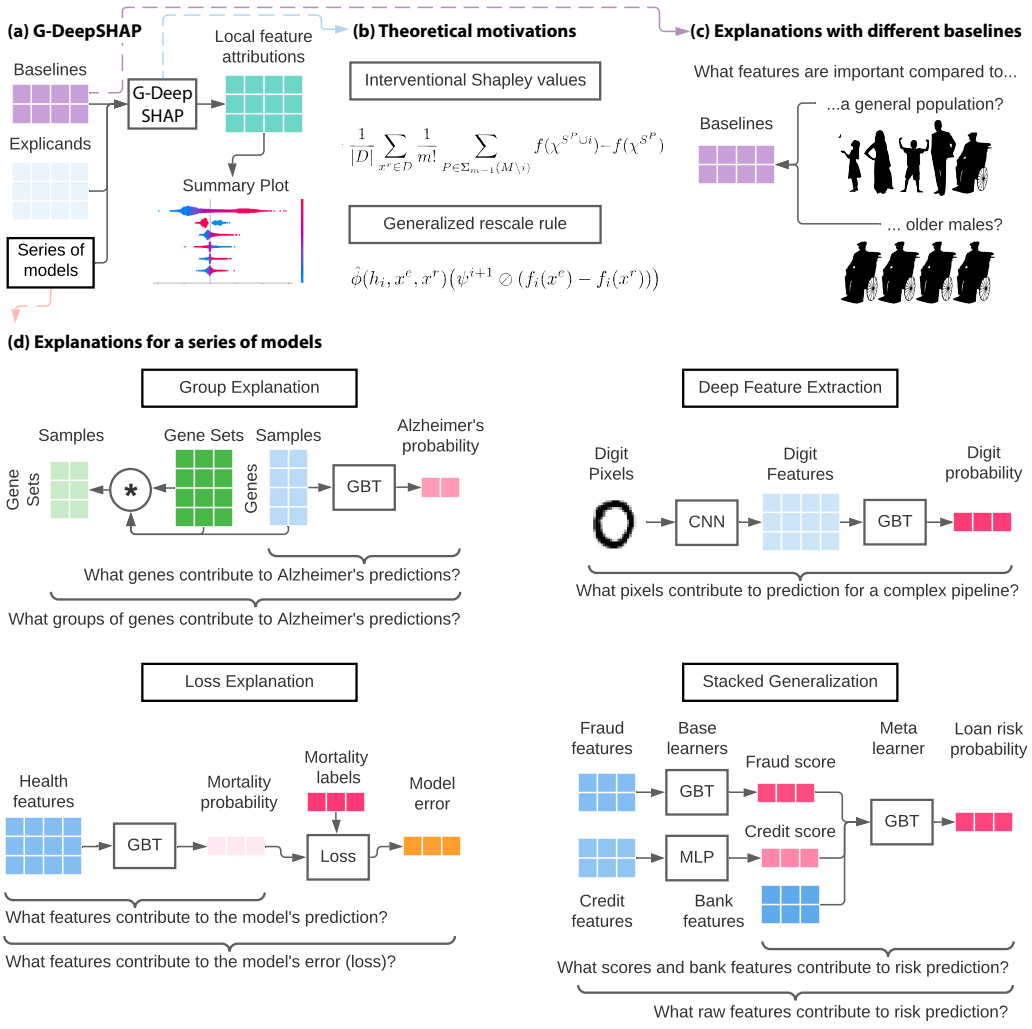


Figure 6.1: G-DeepSHAP estimates Shapley value feature attributions to explain a series of models using a baseline distribution. (a) Local feature attributions with G-DeepSHAP require explicands (samples being explained), a baseline distribution (samples being compared to), and a model that is comprised of a series of models. They can be visualized to understand model behavior (Supplementary Methods Section 1.3). (b) Theoretical motivation behind G-DeepSHAP (Section 6.5.1 and Section 6.5.4). (c) The baseline distribution is an important, but often overlooked, parameter that changes the scientific question implicit in the local feature attributions we obtain. (d) Explaining a series of models enables us to explain groups of features, model loss, and complex pipelines of models (deep feature extraction and stacked generalization). Experimental setups are described in Supplementary Methods Section 1.2.

faster than model-agnostic approaches, but generally cannot be used to explain a series of models. Examples include those for (1) deep models (e.g., DeepLIFT [175], Integrated Gradients [189]) and (2) tree models (e.g., Gain/Gini Importance [23], TreeSHAP [123]).

In this paper, we present Generalized DeepSHAP (G-DeepSHAP) – a local feature attribution method that is faster than model-agnostic methods and can explain complex series of models that pre-existing model-specific methods cannot. G-DeepSHAP is based on connections to the Shapley value, a concept from game theory that satisfies many desirable axioms. We make several important contributions:

1. We propose a theoretical framework (Section 6.5.6) that connects the rules introduced in Shrikumar et al. to the Shapley value with an marginal expectation set function (marginal Shapley value) (Section 6.5.1).

2. We show that the ICE Shapley value decomposes into an average over *single baseline attributions* (Section 6.5.2), where a single baseline attribution explains the model for a single sample (explicand) by comparing to a single sample (baseline).

3. We propose a *generalized rescale rule* to explain a complex series of models by propagating attributions while enforcing efficiency at each layer (Figure 6.1b, Section 6.5.4). This framework extends DeepSHAP to explain any series of models composed of linear, deep, and tree models.

4. We propose a *group rescale rule* to propagate local feature attributions to groups of features (Section 6.5.7). We show that these group attributions better explain models with many features.

Many feature attribution methods must define the absence of a feature, often by masking features according to a single baseline sample (single baseline attribution) [175, 189, 188]. In contrast, we show that under certain assumptions, the correct approach is to use many baseline samples instead (Supplementary Method Section 1.5.3). Qualitatively, we show that

using many baselines avoids bias that can be introduced by single baseline attributions (Section 6.3.1). Additionally, we show that the choice of baseline samples is a useful parameter which changes the question answered by the attributions (Figure 6.1c, Section 6.3.2).

We qualitatively and quantitatively evaluate G-DeepSHAP in real-world datasets including biological, health, image, and financial data sets. In the biological datasets [3, 17, 46, 151], we qualitatively assess group feature attributions based on gene sets identified in prior literature (Section 6.3.3). In the health, image, and financial datasets [44, 111, 1], we quantitively show that G-DeepSHAP provides useful explanations and is drastically faster than model agnostic approaches using an ablation test, where we hide features according to their attribution values (Section 6.3.4, Section 6.3.5, Section 6.3.6). We compare to extremely popular model-agnostic methods including KernelSHAP and IME which are unbiased stochastic estimators for the Shapley value [183, 121, 40] (Supplementary Methods Section 1.5.9).

In practice, G-DeepSHAP can use feature attributions to ask many important scientific questions by explaining different parts of the series of models (Figure 6.1d). When features used by upstream models are semantically meaningless (deep feature extraction) or hard to understand (stacked generalization), G-DeepSHAP provides explanations in terms of the original features which can often be more intuitive, especially for non-technical consumers. In addition, G-DeepSHAP enables attributions with respect to different aspects of model behavior such as predicted risk or even errors the model makes (loss explanation). Finally, using the group rescale rule enables users to reduce the dimensionality of highly correlated features which makes them easier to understand (group explanation).

In addition, G-DeepSHAP is the only approach we are aware of that enables explanations of a distributed series of models (where each model belongs to a separate institution). Model-agnostic approaches do not work because they need access to every model in the series, but institutions cannot share models because they are proprietary. One extremely prevalent example of distributed models are *consumer scores* which exist for nearly every American consumer [51] (Section 6.3.6). In this setting, transparency is a critical issue, because opaque scores can hide discrimination or unfair practices.

A preliminary version of this manuscript appeared at a workshop, entitled "Explaining Models by Propagating Shapley Values of Local Components" [30].

6.2 G-DeepSHAP

In this paper, we improve upon two previous approaches (DeepLIFT [175], DeepSHAP [121]) that propagate attributions while maintaining efficiency with respect to a single baseline. We make two improvements: (1) we compare to a distribution of baselines, which decreases the reliance of the attributions on any single baseline (Section 6.3.1) and (2) we generalize the rescale rule so that it applies to series of mixed model types, rather than only layers in a deep model.

More precisely, a closely related method named DeepSHAP was designed to explain deep models ($f : \mathbb{R}^m \to \mathbb{R}$) [121], by performing DeepLIFT [175] using the average as a baseline [121] (Section 6.5.8). However, using a single average baseline is not the correct approach to explain non-linear models based on connections to Shapley values with an interventional conditional (marginal) expectation set function and a flat causal graph (i.e., a causal graph where arrows are only drawn between input variables and the output) [87]. Instead, we show that the correct way to obtain the marginal Shapley value local feature attributions (denoted as $\phi(f, x^e) \in \mathbb{R}^m$) based on an explicand ($x^e \in \mathbb{R}^m$), or sample being explained, is to average over single baseline feature attributions (denoted as $\phi(f, x^e, x^b) \in \mathbb{R}^m$) where baselines are $x^b \in \mathbb{R}^m$ and D is the set of all baselines (details in Section 6.5.2):

$$\phi(f, x^e) = \frac{1}{|D|} \sum_{x^b \in D} \phi(f, x^e, x^b) \tag{6.1}$$

DeepLIFT [175] explains deep models by propagating feature attributions at each layer of the deep model. Here, we extend DeepLIFT by generalizing DeepLIFT's rescale rule to accommodate more than neural network layers while guaranteeing layer-wise efficiency (details in Section 6.5.4). For a series of models which can be represented as a composition of functions ($f_k(x) = (h_k \circ \cdots \circ h_1)(x)$, where $h_i : \mathbb{R}^{m_i} \to \mathbb{R}^{o_i}$, $m_i = o_{i-1} \forall i \in 2, \cdots k$, $m_1 = m$,

and $o_k = 1$) with intermediary models ($f_i(x) = (h_i \circ \cdots \circ h_1)(x)$). In words, m_i are the input dimensions and o_i are the output dimensions for each layer i. G-DeepSHAP attributions are computed as:

$$\psi^k = \hat{\phi}(h_k, x^e, x^b) \tag{6.2}$$

$$\psi^i = \hat{\phi}(h_i, x^e, x^b)\big(\psi^{i+1} \oslash (f_i(x^e) - f_i(x^b))\big),\ i \in 1, \cdots, k-1. \tag{6.3}$$

We use Hadamard division to denote an element-wise division of $\vec{a}$ by $\vec{b}$ that accommodates zero division, where, if the denominator b_i is 0, we set a_i/b_i to 0. The attributions $\hat{\phi}$ for a particular model in the stack are computed utilizing DeepLIFT with the rescale rule for deep models [175], interventional TreeSHAP for tree models [123], or exactly for linear models. Each intermediate attribution ψ^i serves as feature attribution that satisfies efficiency for h_i's input features, where the attribution in the raw feature space is given by ψ^1. This approach takes inspiration from the chain rule applied specifically to deep networks in [175], that we extend to more general classes of models.

G-DeepSHAP is an approximate method, meaning that it is biased for the true marginal Shapley values (Supplementary Notes Section 2.11). However, this bias allows G-DeepSHAP to be drastically faster than alternative approaches. This strategy of trading bias for speed is taken by other Shapley value estimators including L-Shapley [34], C-Shapley [34], Deep Approximate Shapley Propagation [9], and Shapley Explanation Networks [202] (Supplementary Methods Section 1.5). To ensure the attributions are valuable despite this bias, we extensively evaluate G-DeepSHAP both qualitatively and quantitatively in the following sections.

6.3 Results

6.3.1 Baseline distributions avoid bias

We now use G-DeepSHAP to explain deep models with different choices of baseline distributions to empirically evaluate our theoretical connections to marginal expectations. We show

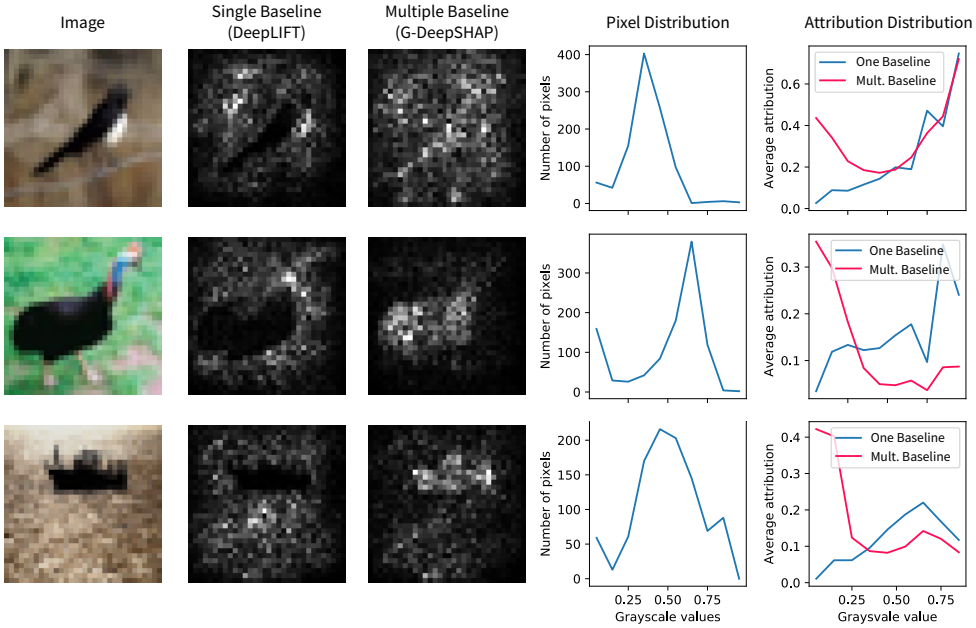


Figure 6.2: Using a single all-black baseline image (DeepLIFT) leads to biased attributions compared to attributions with a randomly sampled baseline distribution (G-DeepSHAP). The image is the explicand. The attribution plots are the sum of the absolute value of the feature attributions for the three channels of the input image. The pixel distribution is the distribution of pixels in terms of their grayscale values. The attribution distribution is the amount of attribution mass upon a group of pixels binned by their grayscale values.

that single baseline attributions are biased in a CNN that achieves 75.56% test accuracy (hyperparameters in Supplementary Methods Section 1.2.1) in the CIFAR10 data set [107]. We aim to demonstrate that single baselines can lead to bias in explanations by comparing attributions using either a single baseline (an all-black image) as in DeepLIFT or a random set of 1000 baselines (random training images) as in G-DeepSHAP. Although the black pixels in the image are qualitatively important, using a single baseline leads to biased attributions with little attribution mass for black pixels (Figure 6.2). In comparison, averaging over multiple baselines leads to qualitatively more sensible attributions. Quantitatively, we show that despite the prevalence of darker pixels (pixel distribution plots in Figure 6.2), single baseline attributions are biased to give them low attribution, whereas averaging over many baselines more sensibly assigns a large amount of credit to dark pixels (attribution distribution plots in

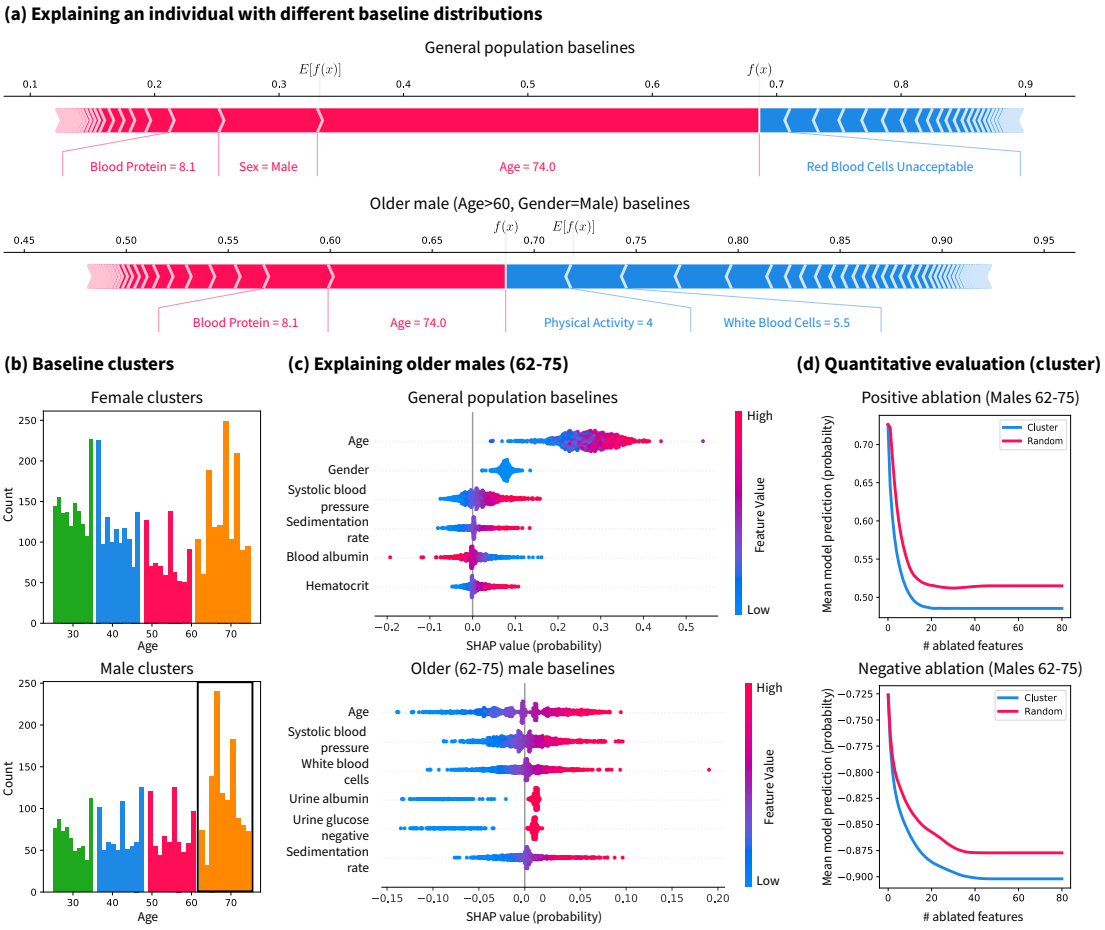


Figure 6.3: The baseline distribution is an important parameter for model explanation. (a) Explaining an older male explicand with both a general population baseline distribution and an older male baseline distribution. Red colors denote positive attributions and blue denote negative attributions. (b) Automatically finding baseline distributions using 8-means clustering on age and gender. Each cluster is shown in a different color. (c) Explaining the older male subpopulation (62-75 years old) with either a general population baseline or an older male baseline. (d) Quantitative evaluation of the feature attributions via positive and negative ablation tests where we mask with the mean of the older male subpopulation (the negative ablation test reports negative mean model output so that lower is better). Note that (b) shows summary plots (Supplementary Methods Section 1.3.3) and (c) shows dependence plots (Supplementary Methods Section 1.3.2).

Figure 6.2). To generalize this finding beyond G-DeepSHAP, we replicate this bias for IME and IG, two popular feature attribution methods that similarly rely on baseline distributions (Supplementary Notes Section 2.1).

6.3.2 Natural scientific questions with baseline distributions

To demonstrate the importance of baseline distributions as a parameter, we explain an MLP (hyperparameters in Supplementary Methods Section 1.2.2) with 0.872 ROC AUC for predicting fifteen year mortality in the NHANES I data set. We use G-DeepSHAP to explain an explicand relative to a baseline distribution drawn uniformly from all samples (Figure 6.3a (top)). This explanation places substantial emphasis on age and gender because it compares the explicand to a population that includes many younger/female individuals. However, in practice epidemiologists are unlikely to compare a 74-year old male to the general population. Therefore, we can manually select a baseline distribution of older males to reveal novel insights, as in Figure 6.3a (bottom). The impact of gender is gone because we compare only to males, and the impact of age is lower because we compare only to older individuals. Furthermore, the impact of physical activity is much higher possibly because physical activity increases active life expectancy, particularly in older populations [37]. This example illustrates that the baseline distribution is an important parameter for feature attributions.

To provide a more principled approach to choosing the baseline distribution parameter, we propose k-means clustering to select a baseline distribution (detail in Section 6.5.3). Previous work analyzed clustering in the attribution space or contrasting to negatively/positively labelled samples [132]. In Figure 6.3b, we show clusters according to age and gender. Then, we explain many older male explicands using either a general population or an older male population baseline distribution (Figure 6.3c). When we compare to the older male baselines, the importance of age is centered around zero, gender is no longer important, and the importance orderings of remaining features change. Further, the inquiry we make changes from "What features are important for older males relative to a general population?" to "What features are important for older males relative to other older males?". To quantitatively evaluate whether our attributions answer the second inquiry, we can ablate features in order of their positive/negative importance by masking with the mean of the older male baseline

distribution (Figure 6.3d, (Section 6.5.10)). In both plots, lower curves indicate attributions that better estimated positive and negative importance. For both tests, attributions with a baseline distribution chosen by k-means clustering substantially outperforms a baseline distribution drawn from the general population.

We find that our clustering-based approach to selecting a baseline distribution has a number of advantages. Our recommendation is to choose baseline distributions by clustering according to non-modifiable, yet meaningful, features like age and gender. This yields explanations that answer questions relative to inherently interpretable subpopulations (e.g., older males). The first advantage is that choosing baseline distributions in this way decreases variance in the features that determined the clusters and subsequently reduces their importance to the model. This is desirable for age and gender because individuals typically cannot modify their age or gender in order to reduce their mortality risk. Second, this approach could potentially reduce model evaluation on off-manifold samples when computing Shapley values [108, 66] by considering only baselines within a reasonable subpopulation. The final advantage is that the flexibility of choosing a baseline distribution allows feature attributions to answer natural contrastive scientific questions [132] that improve model comprehensibility, as in Figure 6.3c.

DeepSHAP (and DeepLIFT) have been shown to be very fast and performant explanation methods for explaining deep models [175, 170, 178]. In the following sections, we instead focus on evaluating our extension of DeepSHAP (G-DeepSHAP) to accommodate a series of mixed models (trees, neural networks, and linear models) and address four impactful applications.

6.3.3 Group attributions identify meaningful gene sets

We explain two MLPs trained to predict Alzheimer's disease status and breast cancer tumor stage from gene expression data with test ROC AUC of 0.959 and 0.932, respectively. We aim to demonstrate that our approach to propagating attributions to groups contributes to model interpretability by validating our discoveries with scientific literature. Gene expression

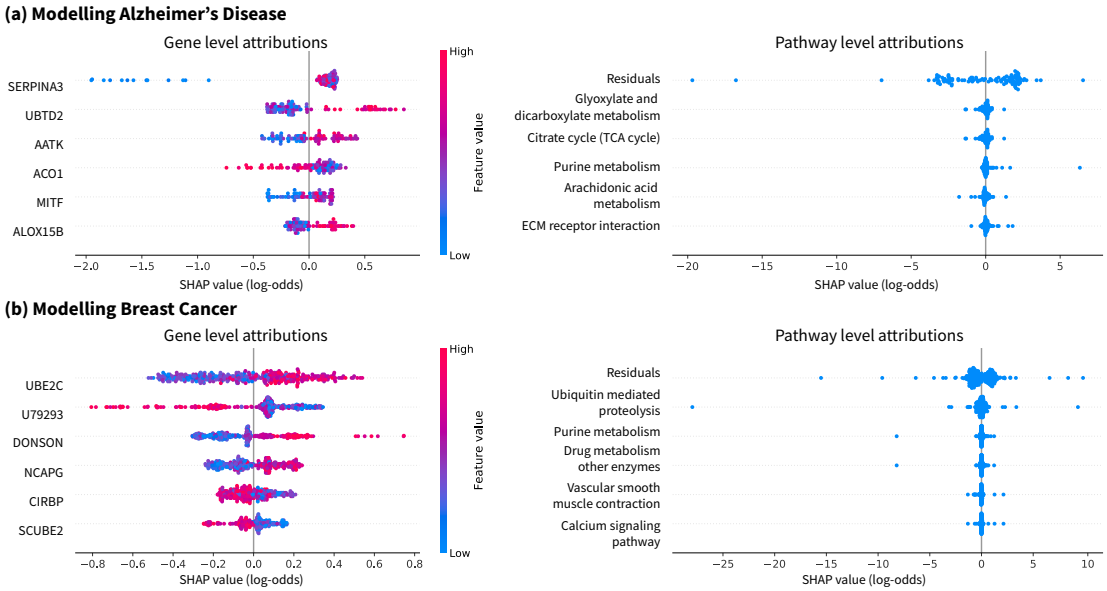


Figure 6.4: Propagating attributions to gene sets enables higher level understanding. (a) Gene and gene set attributions for predicting Alzheimer's disease using gene expression data. (b) Gene and gene set attributions for predicting breast cancer tumor stage using gene expression data. Residuals in the gene set attributions summarize contributions for genes that are not present in any gene set and describes variations in output not described by the pathways we analyzed. Note that (a) and (b) show summary plots (Supplementary Methods Section 1.3.3).

data is often extremely high dimensional; as such, solutions such as gene set enrichment analysis (GSEA) are widely used [187]. In contrast, we aim to attribute importance to gene sets while maintaining efficiency by proposing a *group rescale rule* (Section 6.5.7). This rule sums attributions for genes belonging to each group and then normalizes according to excess attribution mass due to multiple groups containing the same gene. It generalizes to arbitrary groups of features beyond gene sets, such as categories of epidemiological features (e.g., laboratory measurements, demographic measurements, etc.).

In Figure 6.4, we can validate several key genes identified by G-DeepSHAP. For Alzheimer's disease, the overexpression of SERPINA3 has been closely tied to prion diseases [199], and UBTD2 has been connected to frontotemporal dementia – a neurodegenerative disorder [193]. For breast cancer tumor stage, UBE2C was positively correlated with tumor size and his-

tological grade [136]. In addition to understanding gene importance, understanding higher level importance can be obtained using gene sets, i.e., groups of genes defined by biological pathways or co-expression. We obtain gene set attributions by grouping genes according to curated gene sets from the KEGG (Kyoto Encyclopedia of Genes and Genomes) pathway database (https://www.gsea-msigdb.org/gsea/msigdb/collections.jsp#C2) (additional gene set attributions in Supplementary Notes Section 2.5)

Next, we verify important gene sets identified by G-DeepSHAP. For Alzheimer's disease, the glyoxylate and dicarboxylate metabolism pathway was independently identified based on metabolic biomarkers [214]; several studies have demonstrated aberrations in the TCA cycle in Alzheimer's disease brain [10]; and alterations of purine-related metabolites are known to occur in early stages of Alzheimer's disease [8]. For breast cancer, many relevant proteins are involved in ubiquitin-proteasome pathways [145] and purine metabolism was identified as a major metabolic pathway differentiating a highly metastatic breast cancer cell line from a slightly metastatic one [101]. Identifying these phenotypically relevant biological pathways demonstrates that our group rescale rule identifies important pathways.

6.3.4 Loss attributions provide insights to model behavior

We examine an NHANES (1999-2014) mortality prediction GBT model (0.868 test set ROC AUC) to show how explaining the model's loss (loss explanations) provides important insights different from insights revealed by explaining the model's output (output explanations). G-DeepSHAP lets us explain transformations of the model's output. For instance, we can explain a binary classification model in terms of its log-odds predictions, its probability predictions (often easier for non-technical collaborators to understand; see Supplementary Notes Section 2.4), or its loss computed based on the prediction. Here, we focus on local feature attributions that explain per-sample loss.

We train our model on the first five release cycles of the NHANES data (1999-2008) and evaluate it on a test set of the last three release cycles (2009-2014) (Figure 6.5a). As a motivating example, we simulate a covariate shift in the weight variable by re-coding it

to be measured in pounds, rather than kilograms, in release cycles 7 and 8 (Figure 6.5b). Then, we ask, "Can we identify the impact of the covariate shift with feature attributions?" Comparing the train and test output attributions, release cycles 7 and 8 are skewed, but they mimic the same general shape of the training set attributions. If we did not color by release cycles, it might be difficult to identify the covariate shift. In contrast, for loss attributions with positive labels, we can identify that the falsely increased weight leads to many misclassified samples where the loss weight attribution exceeds the expected loss. Although such debugging is powerful, it is not perfect. Note that in the negatively labelled samples, we cannot clearly identify the covariate shift because higher weights are protective and lead to more confident negative mortality prediction.

Next, we examine the natural generalization gap induced by covariate shift over time, which shows a dramatically different loss in the train and test sets (Figure 6.5c). We can see that output attributions are similarly shaped between the train and test distributions; however, the loss attributions in the test set are much higher than in the training set. We can quantitatively verify that negative blood lead affects model performance more in the test set by ablating blood lead for the top 10 samples in the train and test sets according to their loss distributions. From this, we can see that blood lead constitutes a substantial covariate shift in the model's loss and helps explain the observed generalization gap.

As an extension of the quantitative evaluation in Figure 6.5c, we can visualize the impact on the model's loss of ablating by output attributions compared to ablating by loss attributions (Figure 6.5d). This ablation test (Section 6.5.10) asks "What features are important to the model's performance (loss)?" Ablating the positive and negative attributions both increase the mean model loss by hiding features central to making predictions. However, ablating by the negative loss attribution directly increases the loss far more drastically than ablating by the output. More so, ablating positive loss attributions clearly decreases the mean loss, which is not achievable by output attribution ablation. Finally, we compare loss attributions computed using either a model-agnostic approach or G-DeepSHAP. In this setting, G-DeepSHAP is two orders of magnitude faster than model-agnostic approaches

(IME, KernelSHAP, and LIME) while showing extremely competitive positive loss ablation performance and the best negative loss ablation performance.

6.3.5 Explaining deep image feature extractors

We compare G-DeepSHAP explanations to a number of model-agnostic explanations for a series of two models: a CNN feature extractor fed into a GBT model that classifies MNIST zeros with 0.998 test accuracy. In this example, non-linear transformations of the original feature space improve performance of the downstream model (Supplementary Notes Section 2.6) but make model-specific attributions impossible. Qualitatively, we can see that G-DeepSHAP and IME are similar, whereas KernelSHAP is similar for certain explicands but not others (Figure 6.6a). Finally, LIME's attributions show the shape of the original digit, but there is a consistent attribution mass around the surrounding parts of the digit. Qualitatively, we observe that the G-DeepSHAP attributions are sensible. The pixels that constitute the zero digit and the absence of pixels in the center of the zero are important for a positive zero classification.

In terms of quantitative evaluations, we report the runtime and performance of the different approaches in Figure 6.6b. We see that G-DeepSHAP is an order of magnitude faster than model-agnostic approaches, with KernelSHAP being the second fastest. Then, we ablate the top 10% of important positive or negative pixels to see how the model's prediction changes. If we ablate positive pixels, we would expect the model's predictions to drop, and vice versa for negative pixels; doing both showed that G-DeepSHAP outperforms KernelSHAP and LIME, and performs comparably to IME at greatly reduced computational cost.

6.3.6 Explaining distributed proprietary models

We evaluate G-DeepSHAP explanations for a consumer scoring example that feeds a simulated GBT fraud score model and a simulated MLP credit score model into a GBT bank model, which classifies good risk performance (0.681 test ROCAUC) (Figure 6.7). Consumer

scores (e.g., credit scores, fraud scores, health risk scores, etc.) describe individual behavior with predictive models [51]. A vast industry of data brokers generates consumer scores based on a plethora of consumer data. For instance, a single data broker in a 2014 FTC study had 3000 data segments on nearly every consumer in the United States, and another broker added three billion new records to its databases each month [169]. As an example of this, the HELOC data set had an ExternalRiskEstimate feature that we removed because it was opaque. Unfortunately, explaining the models that use consumer scores can obscure important features. For instance, explaining the bank model in Figure 6.7a will tell us that fraud and credit scores are important (in Figure 6.7c), but these scores are inherently opaque to consumers [51]. The truly important features may instead be those that these scores use. A better solution might be model-agnostic methods that explain the entire pipeline at once. However, the model-agnostic approaches require access to all models. In Figure 6.7a, a single institution would have to obtain access to fraud, credit, and bank models to use the standard model-agnostic approaches (Figure 6.7b (left)). This may be fundamentally impractical because each of these models is proprietary. This opacity is concerning given the growing desire for transparency in artificial intelligence [73, 51, 169].

G-DeepSHAP naturally addresses this obstacle by enabling attributions to the original features without forcing companies to share their proprietary models if each institution in the pipeline agrees to work together and has a consistent set of baselines. Furthermore, G-DeepSHAP can combine any other efficiency-satisfying feature attribution method in an analogous way (e.g., integrated/expected gradients [189]). Altogether, G-DeepSHAP constitutes an effective way to glue together explanations across distributed models in industry. In particular, in Figure 6.7a, the lending institution can explain its bank model in terms of bank features and fraud and credit scores. The bank then sends fraud and credit score attributions to their respective companies, who can use them to generate G-DeepSHAP attributions to the original fraud and credit features. The fraud and credit institutions then send the attributions back to the bank, which can provide explanations in terms of the original, more interpretable features to their applicants (Figure 6.7d).

We first quantitatively verify that the G-DeepSHAP attributions for this pipeline are comparable to the model agnostic approaches in Figure 6.7b. We once again see that G-DeepSHAP attributions are competitive with the best performing attributions methods for ablating the top 5 most important positive or negative features. Furthermore, we see that G-DeepSHAP is several orders of magnitude faster than the best performing ablation methods (KernelSHAP and IME) and an order of magnitude faster and much more performant than LIME.

We can qualitatively verify the attributions in Figure 6.7c-d. In Figure 6.7c, we find that the fraud and credit scores are extremely important to the final prediction. In addition, bank features including low revolving balance divided by credit limit (NetFractionRevolvingBurden) and low number of months since inquisitions (MSinceMostRecentInqExcl7Days) are congruously important to good risk performance. Then, in Figure 6.7d we use the generalized rescale rule to obtain attributions in the original feature space. Doing so uncovers important variables hidden by the fraud and credit scores. In particular, we see that the fraud score heavily relied on a high number of months since the applicants's oldest trade (MSinceOldestTradeOpen), and the credit score relied on a low number of months since recent delinquency (MSinceMostRecentDelq) in order to identify applicants that likely had good risk performance. Importantly, the pipeline we analyze in Figure 6.7a also constitutes a stacked generalization ensemble, which we analyze more generally in Supplementary Notes Section 2.7.

6.4 Discussion

In this manuscript, we presented examples where explaining a series of models is critical. Series of models are prevalent in a variety of applications (health, finance, environmental science, etc.), where understanding model behavior contributes important insights. Furthermore, having a fast approach to explain these complex pipelines may be a major desiderata for a diagnostic tool to debug ML models.

The practical applications we focus on in this paper include gene set attribution, where

the number of features far surpasses the number of samples. In this case, we provide a rule that aggregates group attributions to higher level groups of features while maintaining efficiency. Second, we demonstrate the utility of explaining transformations of a model's default output (Supplementary Notes Section 2.4). Explaining the probability output rather than the log-odds output of a logistic model yields more naturally interpretable feature attributions. Furthermore, explaining the loss of a logistic model enables debugging model performance and identification of covariate shift. A third application is neural network feature extraction, where pipelines may include transformations of the original features fed into a different model. In this setting we demonstrate the computational tractability of G-DeepSHAP compared to model-agnostic approaches. Finally, because our approach propagates feature attributions through a series of models while satisfying efficiency at each step (Section 6.5.5), the intermediary attributions at each part of the network can be interpreted as well. We use this to understand the importance of both consumer scores and the original features used by the consumer scores.

In consumer scoring, distributed proprietary models (i.e., models that exist in different institutions) have historically been an obstacle to transparency. This lack of transparency is particularly concerning given the prevalence of consumer scores, with some data brokers having thousands of data segments on nearly every American consumer [169]. In addition, many new consumer scores fall outside the scope of previous regulations (e.g., the Fair Credit Reporting Act and the Equal Credit Opportunity Act) [51]. In fact, these new consumer scores that depend on features correlated with protected factors (e.g., race) can reintroduce discrimination hidden behind proprietary models, which is an issue that has historically been a concern in credit scores (the oldest existing example of a consumer score) [51]. G-DeepSHAP naturally enables feature attributions in this setting and takes a significant and practical step towards increasing the transparency of consumer scores and provides a tool to help safeguard against hidden discrimination.

It should be noted that we focus specifically on evaluating G-DeepSHAP for a series of mixed model types. Previous work evaluates the rescale rule for explaining deep models,

specifically. The original presentation of the rescale rule [175] demonstrates its applicability to deep networks in explaining digit classification and regulatory DNA classification. Schwab and Karlen show that for explaining deep networks, G-DeepSHAP, which uses multiple baselines, is a very fast yet performant approach in terms of an ablation test for explaining MNIST and CIFAR images. Although their approach, CXPlain, is comparably fast at attribution time, it has the added cost of training a separate explanation model. Finally, Sixt et al. shows that many modified back propagation feature attribution techniques are independent of the parameters of later layers, with the exception of DeepLIFT. This particularly significant finding suggests that compared to most fast back propagation-based deep feature attribution approaches, approaches based on the rescale rule are not ignorant of later layers in the network.

Although G-DeepSHAP works very well for explaining a series of mixed model types in practice, an inherent limitation is that it is not guaranteed to satisfy the desirable axioms (e.g., implementation invariance) that other feature attribution approaches satisfy (assuming exact solutions to their intractable problem formulations) [183, 121, 189]. This suggests that G-DeepSHAP may be more appropriate for model debugging or for identifying scientific insights that warrant deeper investigation, particularly in settings where models or the input dimension is huge and tractability is a major concern. However, for applications where high-stakes decision making is important, it may be more appropriate to run axiomatic approaches to completion or use interpretable models [165]. Furthermore, in many real world circumstances, such as distributed proprietary models based on credit risk scores, exact axiomatic approaches and interpretable models are not feasible. In these cases G-DeepSHAP represents a promising direction that allows multiple agents to collaboratively build explanations while maintaining separation of model ownership.

6.5 Methods

We include detailed descriptions of data sets in Supplementary Methods Section 1.1, experimental setups in Supplementary Methods Section 1.2, and feature attribution plots in

Supplementary Methods Section 1.3.

6.5.1 The Shapley value

The Shapley value is a solution concept for allocating credit among players ($M = 1, \cdots, m$) in an m-person game. The game is fully described by a set function $v(S) : \mathcal{P}(S) \to \mathbb{R}^1$ that maps the power set of players $S \subseteq M$ to a scalar value. The Shapley value for player i is the average marginal contribution of that player for all possible permutations of remaining players:

$$\phi_i(v) = \frac{1}{m!} \sum_{P \in \Sigma_{m-1}(M)} (v(S^P \cup i) - v(S^P)). \tag{6.4}$$

We denote the finite symmetric group $\Sigma_{n-1}(M)$, which is the set of all possible permutations, and S^P to be the set of players before player i in the permutation P. The Shapley value is a provably unique solution under a set of axioms (Supplementary Methods Section 1.4). One axiom that we focus on in this paper is *efficiency*:

$$\sum_{i=1}^{m} \phi_i(v) = v(M) - v(\emptyset). \tag{6.5}$$

Adapting the Shapley value for feature attribution of ML models

Unfortunately, the Shapley value cannot assign credit for an ML model ($f(x) : \mathbb{R}^m \to \mathbb{R}^1$) directly because most models require inputs with values for every feature, rather than a subset of features. Accordingly, feature attribution approaches based on the Shapley value define a new set function $v(S)$ that is a *lift* of the original model [133]. In this paper, we focus on local feature attributions, which describe a model's behavior for a single sample, called an explicand (x^e). A lift is defined as:

$$\mu(f, x^e, S) : \mathbb{R}^m \times 2^m \to \mathbb{R}^1. \tag{6.6}$$

One common lift is the *conditional expectation*, where the lift is the conditional expecta-

tion of the model's output holding features in S fixed to x_S^e and X is a multivariate random variable with joint distribution D:

$$\mu_D^{obs}(f, x^e, S) = \mathbb{E}_D[f(X)|X_S = x_S^e]. \tag{6.7}$$

Another common lift is the *interventional conditional expectation* with a flat causal graph, where we "intervene" on features by breaking the dependence between features in X_S and the remaining features using the causal inference *do*-operator [87]:

$$\mu_D^{int}(f, x^e, S) = \mathbb{E}_D[f(X)|do(X_S = x_S^e)]. \tag{6.8}$$

Both approaches have tradeoffs that have been described elsewhere [29, 108, 188, 132, 66]. Here, we focus on the marginal approach for two primary reasons:

1. Conditional Shapley values will spread credit among correlated features [29]. Although this can be desirable, it can lead to counterintuitive attributions. In particular, features that the model literally does not use to calculate its predictions will have non-zero attribution simply if they are correlated with features the model heavily depends on [188]. Instead, the marginal Shapley values do a better job of identifying the features the models algebraically depend on [29]. As such, the marginal Shapley values are useful for debugging bad models and drawing insights from good models. In contrast, although conditional Shapley values give a view of the information content each feature has with regard to the output for optimal models [42], this is not the case for bad models. Furthermore, it can be hard to use conditional Shapley values to debug bad models because it is unclear whether a feature is important because it is explicitly depended on by the model or because it is correlated with the features the model explicitly depends on. Finally, if it is really important to spread credit using correlated features, it is possible to modify the model fitting using regularization or ensembles which will cause marginal Shapley values to naturally spread credit [29].

2. Estimating the conditional expectation is drastically harder than the marginal expectation. This is reflected in a wide disagreement about how to estimate the conditional expectation [42], with approaches including empirical [188], cohort refinement [128, 188, 5], parametric assumptions [29, 5], generative model [66], surrogate model [66], missingness during training [42], and separate models [117, 185, 208]. On the other hand, the marginal expectation has one agreed upon empirical estimation strategy [188, 123]. This difficulty also reflects in the model-specific approaches, where there are exact algorithms to calculate marginal Shapley values for linear and tree models (LinearSHAP [29] and TreeSHAP [123]). In particular, TreeSHAP and G-DeepSHAP are both based on the useful property that marginal Shapley values decompose into an average of baseline Shapley values (Section 6.5.2). This benefit is crucial to the design of the generalized rescale rule.

Note that a third approach, named causal Shapley values, uses causal inference's interventional conditional expectation, but does not assume a flat causal graph [80]. Causal Shapley values require knowledge of a causal graph relating the input variables and the output. However, in general this graph is unknown or requires substantial domain expertise. In addition, causal Shapley values are hard to estimate because they require estimating many interventional probabilities. In contrast, marginal Shapley values are a tractable way to understand model behavior.

The Shapley values computed for any lift will satisfy efficiency in terms of the lift μ. However, for the interventional and observational lift described above, the Shapley value will also satisfy efficiency in terms of the model's prediction:

$$\sum_i \phi_i^{\mu_D}(f, x^e) = f(x^e) - \mathbb{E}_D[f(X)]. \tag{6.9}$$

This means that attributions can naturally be understood to be in the scale of the model's predictions (e.g., log-odds or probability for binary classification).

6.5.2 *marginal Shapley values baseline distribution*

We can define a single baseline lift

$$\mu^{int}_{x^b}(f, x^e, S) = \mathbb{E}_{\{x^b\}}[f(X)|do(X_S = x^e_S)] = \chi^S, \tag{6.10}$$

where χ^S is a spliced sample and $\chi^S_i = x^e_i$ if $i \in S$, else $\chi^S_i = x^b_i$.

Then, we can decompose the Shapley value $\phi_i(f, x^e)$ for the interventional conditional expectation lift (Equation (6.8)) (henceforth referred to as the marginal Shapley value) into an average of Shapley values with single baseline lifts (proof in Supplementary Methods Section 1.6):

$$\phi_i(f, x^e, D) = \frac{1}{|D|} \sum_{x^b \in D} \underbrace{\frac{1}{m!} \sum_{P \in \Sigma_{m-1}(M)} f(\chi^{S^P \cup i}) - f(\chi^{S^P})}_{\text{Shapley value for single baseline lift}} \tag{6.11}$$

$$= \frac{1}{|D|} \sum_{x^b \in D} \phi_i(f, x^e, x^b). \tag{6.12}$$

Here, D is an empirical distribution with equal probability for each sample in a baseline data set. An analogous result exists for the conditional distribution lift using an input distribution [132]. The attributions for these single baseline games are also analogous to baseline Shapley in [188].

In the original DeepLIFT paper, [175] recommend two heuristic approaches to define baseline distributions: (1) choosing a sensible single baseline and (2) averaging over multiple baselines. In addition, DeepSHAP, as previously described in [121], created attributions with respect to a single baseline equal to the expected value of the inputs (Section 6.5.8). In this paper, we show that from the perspective of Shapley values with an marginal expectation lift, averaging over feature attributions computed with single baselines drawn from an empirical distribution is the correct approach. One exception to this are linear models, where taking the average as the baseline is equivalent to averaging over many single baseline feature

attributions [29]. marginal Shapley values computed with a single baseline satisfy efficiency in terms of the model's prediction:

$$\sum_i \phi_i(f, x^e, x^b) = f(x^e) - f(x^b). \tag{6.13}$$

6.5.3 Selecting a baseline distribution

As in the previous section, we define a baseline distribution D over which we compute Shapley values with single baseline lifts. This baseline distribution is naturally chosen to be a distribution over the training data X^{train}, where each sample $x^j \in \mathbb{R}^m$ has equal probability. The interpretation of this distribution is that the explicand is compared to each baseline in D. This means that the marginal Shapley values implicitly create attributions that explain the model's output relative to a baseline distribution.

Although the entire training distribution is a natural and interpretable choice of baseline distribution, it may be desirable to use others. To automate the process of choosing such an interpretable baseline distribution, we turn to unsupervised clustering. We utilize k-means clustering on a reduced version of the training data ($\hat{X}^{train}$) comprised of $\hat{x}^j = [x_i^j \forall i \in M_r]$ with a reduced set of features (M_r). The output of the k-means clustering are clusters $C_1, \cdots, C_k$ with means $\mu_1, \cdots, \mu_k$ that minimize the following objective on the reduced training data:

$$\underset{C_1,\cdots,C_k}{\arg\min} \sum_{i=1}^{k} \sum_{\hat{x} \in C_i} ||\hat{x} - \mu_i||^2. \tag{6.14}$$

Then, the cluster selected as a baseline distribution explaining an explicand x^e is chosen based on:

$$\underset{i}{\arg\min}\, ||\hat{x}^e - \mu_i||^2. \tag{6.15}$$

Note that in practice, it is common to use a large subsample of the full baseline distribution. The number of baseline samples can be an important parameter that can be validated by running explanations for multiple replicates and confirming consistency. We evaluate con-

vergence in Supplementary Notes Section 2.3 and find that 1000 baselines lead to consistent attributions.

6.5.4 A generalized rescale rule to explain a series of models

We define a *generalized rescale rule* to explain an arbitrary series of models that propagates approximate Shapley values with an marginal expectation lift for each model in the series. To describe the approach, we define a *series of models* to be a composition of functions $f_k(x) = (h_k \circ \cdots \circ h_1)(x)$, and we define intermediary models $f_i(x) = (h_i \circ \cdots \circ h_1)(x),\ i = 1, \cdots, k$. We define the domain and codomain of each model in the series as $h_i(x) : \mathbb{R}^{m_i} \to \mathbb{R}^{o_i}$. Then, we can define the propagation for a single baseline recursively:

$$\psi^k = \hat{\phi}(h_k, x^e, x^b) \tag{6.16}$$

$$\psi^i = \hat{\phi}(h_i, x^e, x^b)\big(\psi^{i+1} \oslash (f_i(x^e) - f_i(x^b))\big),\ i \in 1, \cdots, k-1. \tag{6.17}$$

We use Hadamard division to denote an element-wise division of $\vec{a}$ by $\vec{b}$ that accommodates zero division, where if the denominator b_i is 0, we set a_i/b_i to 0. Additionally, $\hat{\phi}$ are an appropriate feature attribution technique that approximates marginal Shapley values while crucially satisfying efficiency for the model h_i it is explaining. In this paper, we utilize DeepLIFT (rescale) for deep models, TreeSHAP for tree models, and exact marginal Shapley values for linear models. We define efficiency as $\hat{1}_{1\times m_i}\hat{\phi}(h_i, x^e, x^b) = f_i(x^e) - f_i(x^b)$ where $\hat{1}_{a\times b}$ is a matrix of ones with shape $a \times b$ and the approximate Shapley value functions $\hat{\phi}$ return matrices in $\mathbb{R}^{(m_i \times o_i)}$. The final attributions in the original feature space are:

$$\phi_i(f_k, x^e, x^b) = \psi_i^1. \tag{6.18}$$

Furthermore, this approach yields intermediate attributions that serve as meaningful feature attributions. In particular, ψ^i can be interpreted as the importance of the inputs to the model $(h_k \circ \cdots \circ h_i)$, where the new explicand and baseline are $(h_{i-1} \circ \cdots \circ h_1)(x^e)$ and

$(h_{i-1} \circ \cdots \circ h_1)(x^b)$, respectively. This approach takes inspiration from the chain rule applied specifically for deep networks in [175], but we extend it to more general classes of models.

6.5.5 Efficiency for intermediate attributions

As one might expect, each intermediate attribution ψ^i satisfies efficiency:

Theorem 1: Each attribution $\psi^i \in \mathbb{R}^m, \forall i \in 1, \cdots, k$ satisfies efficiency and sums up to $f_k(x^e) - f_k(x^b)$.

Proof:

We will prove by induction that

$$\hat{1}_{1\times m_i}\psi^i = f_k(x^e) - f_k(x^b), \forall i \in 1, \cdots, k. \tag{6.19}$$

For simplicity of notation, denote $\hat{\phi}^i = \hat{\phi}(h^i, x^e, x^b)$.

Assumption: Each $\hat{\phi}$ satisfies efficiency

$$\hat{1}_{1\times m_i}\hat{\phi}^i = f_i(x^e) - f_i(x^b). \tag{6.20}$$

Base Case: By our assumption,

$$\hat{1}_{1\times m_k}\psi^k = f_k(x^e) - f_k(x^b). \tag{6.21}$$

Induction Step:

$$\psi^i = \hat{\phi}\big(\psi^{i+1} \oslash (f_i(x^e) - f_i(x^b))\big) \tag{6.22}$$

$$\hat{1}_{1\times m_i}\psi^i = \hat{1}_{1\times m_i}\hat{\phi}\big(\psi^{i+1} \oslash (f_i(x^e) - f_i(x^b))\big) \tag{6.23}$$

$$= (f_i(x^e) - f_i(x^b))\big(\psi^{i+1} \oslash (f_i(x^e) - f_i(x^b))\big) \tag{6.24}$$

$$= \hat{1}_{1\times o_i}\psi^{i+1} \tag{6.25}$$

$$= \hat{1}_{1\times m_{i+1}}\psi^{i+1} \tag{6.26}$$

$$= f_k(x^e) - f_k(x^b). \tag{6.27}$$

Conclusion: By the principle of induction, each intermediate attribution satisfies efficiency (Equation (6.19)).

Then, because the marginal Shapley value with a baseline distribution is the average of many single baseline attributions, it satisfies a related notion of efficiency:

$$\sum_i \phi_i(f_k, x^e) = \sum_i \sum_{x^b \in D} \phi_i(f_k, x^e, x^b) \tag{6.28}$$

$$= \sum_{x^b \in D} \sum_i \phi_i(f_k, x^e, x^b) \tag{6.29}$$

$$= \sum_{x^b \in D} f_k(x^e) - f_k(x^b) \tag{6.30}$$

$$= f_k(x^e) - \frac{1}{|D|} \sum_{x^b \in D} f_k(x^b). \tag{6.31}$$

This can be naturally interpreted as the difference between the explicand's prediction and the expected value of the function across the baseline distribution.

An additional property of the generalized rescale rule is that although it is an approximation to the marginal Shapley values in the general case, if every model in the composition is linear ($h_i(x) = \beta x$), then this propagation exactly yields the marginal Shapley values (Supplementary Methods Section 1.7).

6.5.6 Connecting DeepLIFT's rules to the Shapley values

Now we can connect the Shapley values to DeepLIFT's Rescale and RevealCancel rules. Both rules aim to satisfy an efficiency axiom (what they call *summation to delta*) and can be connected to an marginal expectation lift with a single baseline (as in Section 6.5.3). Note that although the Rescale rule does not explicitly account for interaction effects, they can be captured in deep models, which we visualize in Supplementary Notes Section 2.8.

In fact, multi-layer perceptrons are a special case where the models in the series are non-linearities applied to linear functions. We first represent deep models as a composition of functions $(h_1 \circ \cdots \circ h_k)(x)$. The Rescale and RevealCancel rules canonically apply to a specific class of function: $h_i(x) = (f \circ g)(x)$, where f is a non-linear function and g is a linear function parameterized by $\beta \in \mathbb{R}^m$. We can interpret both rules as an approximation to marginal Shapley values based on the following definition.

Definition 6.5.1 (k-partition approximation)**.** *A k-partition approximation to the Shapley values splits the features in $x \in \mathbb{R}^m$ into K disjoint sets. Then, it exactly computes the Shapley value for each set and propagates it linearly to each component of the set.*

The Rescale rule can be described as a 1-partition approximation to the marginal Shapley values for $h_i(x)$, while the RevealCancel rule can be described as a 2-partition approximation that splits according to whether $\beta_i x_i > t$, where the threshold $t = 0$. This k-partition approximation lets us consider alternative variants of the Rescale and RevealCancel rules that incur exponentially larger costs in terms of K and for different choices of thresholds.

6.5.7 Explaining groups of input features

Here, we further generalize the Rescale rule to support groupings of features in the input space. Having such a method can be particularly useful when explaining models with very large numbers of features that are more understandable in higher level groups. One natural example is gene expression data, where the numbers of features is often extremely large.

We introduce a *group rescale rule* that facilitates higher level understanding of feature attributions. It provides a natural way to impose sparsity when explaining sets of correlated features. Sparsity can be desirable when explaining a large number of features [165]. We can define a set of groups $G_1, \cdots, G_o$ whose members are the input features x_i. If each group is disjoint and covers the full set of features, then a natural group attribution that satisfies efficiency is the sum:

$$\phi^0_{G_j}(f, x^e) = \sum_{i \in G_j} \phi_i(f, x^e). \tag{6.32}$$

If the groups are not disjoint or do not cover all input features, then the above attributions do not satisfy efficiency. To address this, we define a residual group G_R that covers all input features not covered by the remaining groups. Then, the new attributions are a rescaled version of Equation (6.32)

$$\phi_{G_j}(f, x^e) = \phi^0_{G_j}(f, x^e) \times \frac{\sum \phi_{G_j}(f, x^e)}{\sum \phi_i(f, x^e)}. \tag{6.33}$$

We can naturally extend this approach to accommodate non-uniform weighting of group elements, although we do not experiment with this in our paper.

6.5.8 Differences to previous approaches

In the original SHAP paper [121], they aim to calculate conditional Shapley values. However, due to the difficulty in estimating conditional expectations, they actually calculate what is later described as marginal Shapley values [87, 29].

DeepSHAP was originally introduced as an adaptation of DeepLIFT in the original SHAP paper [121] designed to make DeepLIFT closer to the marginal Shapley values. However, it is briefly and informally introduced, making it difficult to know exactly what the method entails and how it differs from DeepLIFT. DeepSHAP is the same as DeepLIFT, but with the reference (baseline) values set to the average of the baseline samples. Similarly to DeepLIFT, using an average baseline also leads to bias (Supplementary Section 2.2). In comparison,

DeepLIFT typically sets the baseline to uninformative values, and sets them to zeros for image data (an all-black image).

However, marginal Shapley values are equivalent to an average of baseline Shapley values, but not to baseline Shapley values with the average as a baseline. Due to this interpretation (Section 6.5.2), it is more natural to calculate G-DeepSHAP as the average of many attributions for different baselines. This in turn allows us to formulate a generalized rescale rule which allows us to propagate attributions through pipelines of linear, tree, and deep models for which baseline Shapley values are easy to calculate.

In order to clarify the differences, we explicitly define DeepSHAP as it was originally briefly proposed in [121] and the current version we are proposing. DeepSHAP used the rescale rule with an average baseline. G-DeepSHAP uses the generalized rescale rule and group rescale rule with multiple baselines. In terms of applications, DeepSHAP only applies to deep models, whereas G-DeepSHAP applies to pipelines of linear, tree, and deep models. Finally, the group rescale rule gives us a natural approach to group large numbers of features and thus generate attributions for a much smaller number of groups. This type of sparsity is often helpful for helping humans understand model explanations [118].

6.5.9 Evaluation of explanations

The evaluation of explanations is the topic of many papers [123, 82, 52, 142, 7]. Although there is unlikely to be a single perfect approach to evaluate local feature attributions, we can roughly separate them into two categories: qualitative and quantitative, which typically correspond to plausibility of explanations and fidelity to model behavior respectively.

Qualitative evaluations aim to ensure that relationships between features and the outcome identified by the feature attributions are correct. In general, this requires a priori knowledge of the underlying data generating mechanism. One setting in which this is possible are synthetic evaluations, where the data generating mechanism is fully known. This can be unappealing because methods that work for synthetic data may not work for real data. Instead, another approach is to externally validate with prior literature. In this case,

qualitative evaluations aim to capture some underlying truth about the world that has been independently verified in diverse studies. This type of evaluation simultaneously validates the combination of the model fitting and the feature attribution itself to see whether explanations are plausible. One downside of this approach is that it can be hard to rigorously compare different explanation techniques because the evaluation is inherently qualitative. Furthermore, if the explanations find a previously unobserved relationship it can be hard to verify. For this reason, our qualitative evaluations take place in well-studied domains: mortality epidemiology, Alzheimer's and breast cancer biology, and financial risk assessment.

Then, quantitative evaluations typically aim to ensure that the feature attributions are representative of model behavior. These evaluations are dominated by feature ablation tests which aim to modify the samples in a way that should produce an expected response in the model's output. Since most local feature attributions aim to explain the model's output, it is a natural aspect of model behavior to measure. In contrast to the qualitative evaluations, quantitative evaluations are typically aimed exclusively at the feature attribution and are somewhat independent to the model being explained. These evaluations, while useful, are also imperfect, because a method that succeeds at describing model behavior perfectly will likely be far too complex to provide an explanation that humans can understand [206].

Therefore, we found it important to balance these two types of evaluations within our paper. We provide qualitative assessments for all-cause mortality, Alzheimer's, breast cancer, and loan risk performance. We additionally provide quantitative assessments for all-cause mortality, digits classification, and the loan risk performance data.

6.5.10 Ablation tests

We quantitatively evaluate our feature attribution methods with *ablation tests* [82, 123]. In particular, we rely on a simple yet intuitive ablation test. For a matrix of explicands $X^e \in \mathbb{R}^{n_e,m}$, we can get attributions $\phi(f, X^e) \in \mathbb{R}^{n_e,m}$. The ablation test is defined by three parameters: (1) the feature ordering, (2) an imputation sample $x^b \in \mathbb{R}^m$, and (3) an evaluation metric. Then, the ablation test replaces features one at a time with the baseline's feature

value based on the feature attributions to assess the impact on the evaluation metric. We can iteratively define the ablation test based on modified versions of the original explicands:

$$X^{e,0} = X^e \tag{6.34}$$

$$X^{e,k} = X^e \odot I_k(\phi) + X^b \odot (1 - I_k(\phi)), \forall k \in 1, \cdots, m. \tag{6.35}$$

Note that $X^b \coloneqq [\underbrace{x^b \cdots x^b}_{n_e \text{ elements}}]^T$, $I_k(\phi) \coloneqq I_k(\phi(f, X^e)) = \arg\max_{k,axis=1}(\phi(f, X^e))$, where $\arg\max_{k,axis=1}(G)$ returns an indicator matrix of the same size as G, 1 indicates that the element was in the maximum k elements across a particular axis, and $\odot$ signifies a Hadamard product.

Then, the ablation test measures the mean model output (e.g., the predicted log-odds, predicted probability, the loss, etc.) if we ablate k features to be the average over the predictions for each ablated explicand:

$$\frac{1}{n_e} \sum_{i \in 1, \cdots, n_e} f(X_i^{e,k}). \tag{6.36}$$

Note that for our ablation tests we focus on either the positive or the negative elements of ϕ, since the expected change in model output is clear if we ablate only by positive or negative attributions. Since each sample is ablated independently based on their attributions, this ablation test can be considered a summary of local ablations (Supplementary Notes Section 2.10) for many different explicands.

Ablation tests are a natural approach to test whether feature attributions are correct for a set of explicands. For feature attributions that explain the predicted log-odds, a natural choice of model output for the ablation test is the mean of the log-odds predictions. Then, as we ablate increasing numbers of features, we expect to see the model's output change. When we ablate the most positive features (according to their attributions), the mean model output should decrease substantially. As we ablate additional features, the mean model output should still decrease, but less drastically so. This implies that, for positive

ablations, lower curves imply attributions that better described the model's behavior. In contrast, for negative ablations, as we ablate the most negative features, better attributions will cause the mean model output to increase rapidly and lead to higher curves. As a final note, we demonstrate that estimates of marginal Shapley values for random tree and deep models based on TreeSHAP and G-DeepSHAP respectively perform well on ablation tests (Supplementary Notes Section 2.9). This implies that our attributions closely describe the model behavior regardless of its predictive performance.

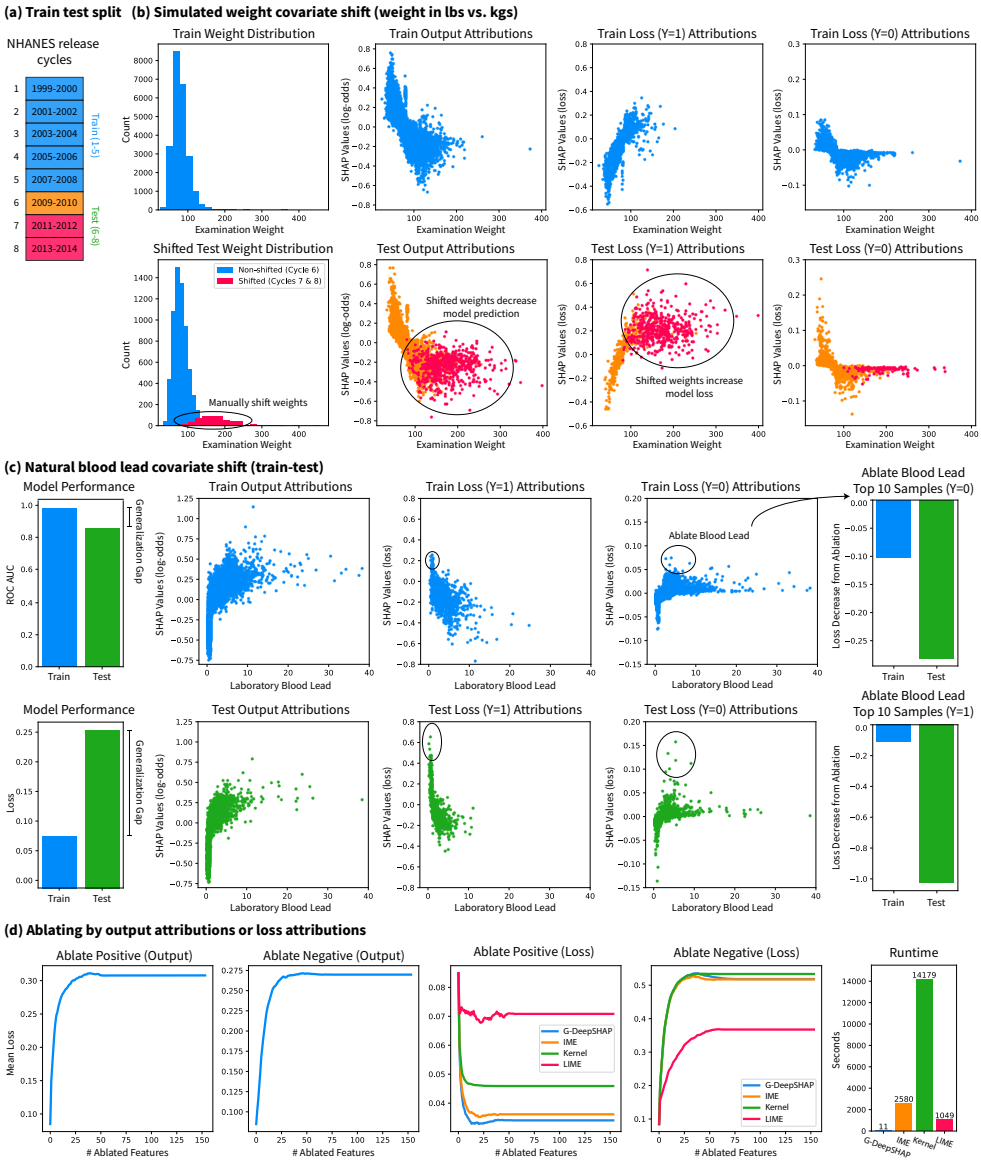


Figure 6.5: Explanations of the model's loss rather than the model's prediction yields new insights. (a) We train on the first five cycles of NHANES (1999-2008) and test on the last three cycles (2009-2014). (b) We identify a simulated covariate shift in cycles 7-8 (2011-2014) by examining loss attributions. (c) Under a natural covariate shift, we identify and quantitatively validate test samples for which blood lead greatly increases the loss in comparison to training samples. (d) We ablate output attributions (G-DeepSHAP) and loss attributions (G-DeepSHAP, IME, KernelSHAP, and LIME) to show their respective impacts on model loss. We compare only to model-agnostic methods for loss attributions because explaining model loss requires explaining a series of models. Note that (b) and (c) show dependence plots (Supplementary Methods Section 1.3.2)).

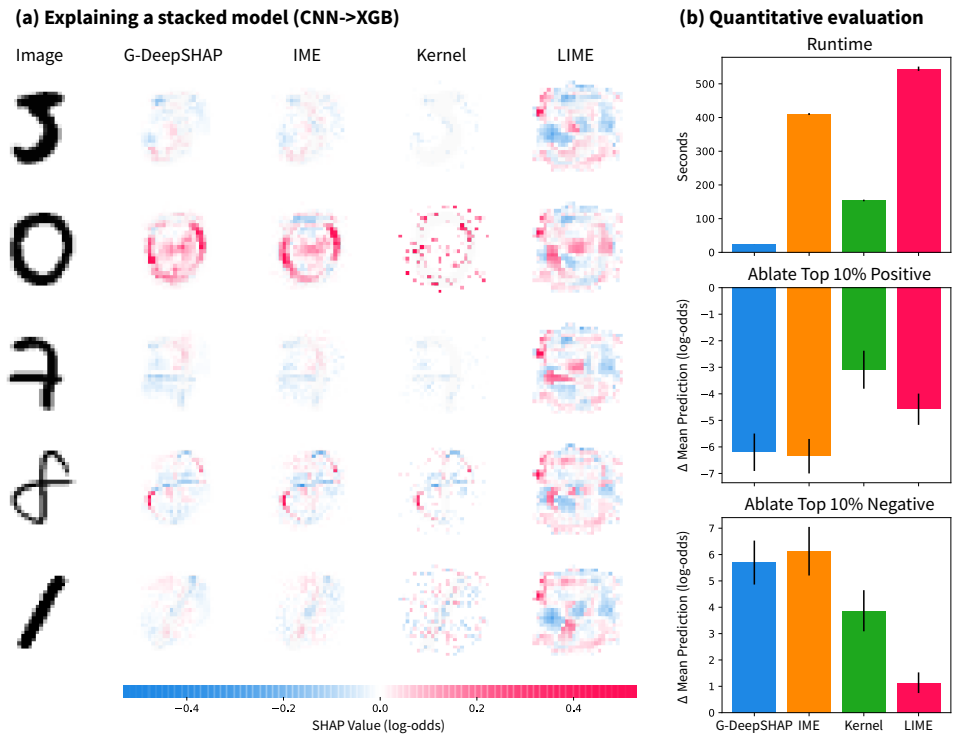


Figure 6.6: Explaining a series of models comprised of a convolutional neural network feature extractor and a gradient boosted tree classifier. (a) Explanations from G-DeepSHAP and state of the art model-agnostic approaches. (b) Quantitative evaluation of approaches, including runtime and ablation of the top 10% of positive and negative features. Error bars are 95% confidence intervals based on twenty iterations of randomly drawing five explicand images, then computing attributions and ablation results.

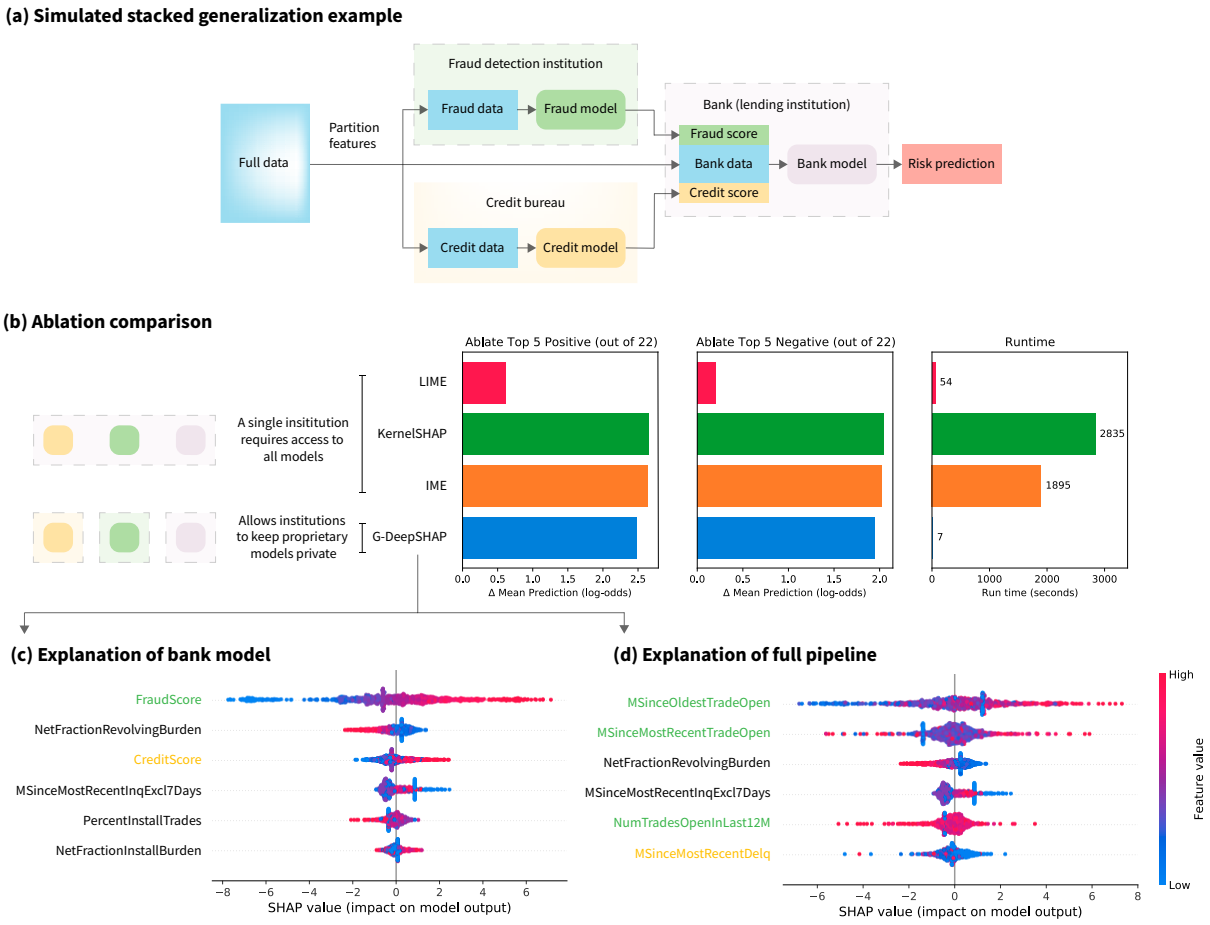


Figure 6.7: Explaining a stacked generalization pipeline of models for the HELOC data set (details in Supplementary Methods Section 1.1.7). (a) A simulated model pipeline in the financial services industry. We partition the original set of features into fraud, credit, and bank features. We train a model to predict risk using fraud data and a model to predict risk using credit data. Then, we use the outputs of the fraud and credit models as scores alongside additional bank features to predict the final customer risk. (b) Ablation tests (ablating top 5 positive/negative features out of a total 22 features) comparing model agnostic approaches (LIME, KernelSHAP, IME), which require access to all models in the pipeline, and G-DeepSHAP, which allows institutions to keep their proprietary models private. (c) Summary plot of the top six features the bank model uses to predict risk (TreeSHAP). (d) Summary plot of the top six features the entire pipeline uses to explain risk (G-DeepSHAP). The green features originate from the fraud data, and the yellow features from the credit data. We explain 1000 randomly samples explicands using 100 randomly sampled baselines for all attribution methods. Note that (c) and (d) show summary plots (Supplementary Methods Section 1.3.3).

Chapter 7

CONCLUSION

The need to explain complex models is rapidly growing in response to the widespread adoption of machine learning models in a variety of fields. The work presented here develops new methods to tractably interpret patterns learned by complex, black box models using Shapley value explanations.

The preceding chapters focus on estimating Shapley value explanations; however, a number of open questions constitute important future research directions in the field of explainable AI.

In terms of Shapley value explanations, important directions include development of new methods and comparisons of existing methods for estimating conditional and causal Shapley values. These variants of Shapley values have nice theoretical properties, but are perhaps fundamentally difficult to estimate. One promising solution is to learn surrogate models to approximate conditional expectations of model outputs, but this comes with a number of difficulties described in Section 2.5.1. In terms of causal Shapley value attributions [80, 202, 67], these approaches are typically difficult to apply because they presuppose the knowledge of a causal graph between features. In image or text settings this may never be available and even for tabular datasets, it can be very difficult to estimate or evaluate causal graphs. However, since causal knowledge is sometimes what users actually want, further development and assessment of techniques to estimate causal graphs is important to Shapley value explanations.

Another challenge which is common to feature attribution methods beyond Shapley value explanations is evaluation. A number of evaluation metrics have been proposed [81, 82, 123, 42], but there are a great variety of metrics and currently no definitive way

to evaluate feature attribution methods. Building on this difficulty of quantitatively evaluating feature attributions, it is perhaps even more important to perform human-centric qualitative evaluations of explanation methods. Qualitative evaluations are difficult because they are expensive and tricky to design, but can be promising strategies to test whether feature attributions are actually useful to outcomes of teams that rely on them [14].

Finally, concept-based explanations [99, 103] are also a particularly promising research direction. Concept-based explanations explain models in terms of higher level concepts, rather than input features, which can be useful if input features are uninterpretable (e.g., due to too many features, strong correlations, etc.). Work in this direction can be very promising to provide explanations which are useful to humans.

BIBLIOGRAPHY

[1] FICO, Xml challenge. https://community.fico.com/s/explainable-machine-learning-challenge. [Online; accessed 01-June-2021].

[2] Otto group product classification challenge. URL https://www.kaggle.com/c/otto-group-product-classification-challenge/discussion/14335.

[3] David A Bennett, Julie A Schneider, Zoe Arvanitakis, and Robert S Wilson. Overview and findings from the religious orders study. *Current Alzheimer Research*, 9(6):628–645, 2012.

[4] Kjersti Aas, Martin Jullum, and Anders Loland. Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *arXiv preprint arXiv:1903.10464*, 2019.

[5] Kjersti Aas, Martin Jullum, and Anders Løland. Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *Artificial Intelligence*, 298:103502, 2021.

[6] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

[7] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

[8] Patricia Alonso-Andres, Jose Luis Albasanz, Isidro Ferrer, and Mairena Martin. Purine-related metabolites and their converting enzymes are altered in frontal, parietal and temporal cortex at early stages of alzheimer's disease pathology. *Brain Pathology*, 28(6):933–946, 2018.

[9] Marco Ancona, Cengiz Oztireli, and Markus Gross. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *International Conference on Machine Learning*, pages 272–281. PMLR, 2019.

[10] Hani Atamna and William H Frey II. Mechanisms of mitochondrial dysfunction and energy deficiency in alzheimer's disease. *Mitochondrion*, 7(5):297–310, 2007.

[11] Robert J Aumann and Lloyd S Shapley. *Values of non-atomic games*. Princeton University Press, 2015.

[12] Robert JJ Aumann. Economic applications of the shapley value. In *Game-theoretic methods in general equilibrium analysis*, pages 121–133. Springer, 1994.

[13] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert MÃžller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010.

[14] Gagan Bansal, Tongshuang Wu, Joyce Zhou, Raymond Fok, Besmira Nushi, Ece Kamar, Marco Tulio Ribeiro, and Daniel Weld. Does the whole exceed its parts? the effect of ai explanations on complementary team performance. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–16, 2021.

[15] Ayman Basali, Edward J Mascha, Iain Kalfas, and Armin Schubert. Relation between perioperative hypertension and intracranial hemorrhage after craniotomy. *Anesthesiology: The Journal of the American Society of Anesthesiologists*, 93(1):48–54, 2000.

[16] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings*

of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6541–6549, 2017.

[17] David A Bennett, Aron S Buchman, Patricia A Boyle, Lisa L Barnes, Robert S Wilson, and Julie A Schneider. Religious orders study and rush memory and aging project. *Journal of Alzheimer's Disease*, 64(s1):S161–S189, 2018.

[18] Samir Bhatt, Ewan Cameron, Seth R Flaxman, Daniel J Weiss, David L Smith, and Peter W Gething. Improved prediction accuracy for disease risk mapping using gaussian process stacked generalization. *Journal of The Royal Society Interface*, 14(134): 20170520, 2017.

[19] Umang Bhatt, Alice Xiang, Shubham Sharma, Adrian Weller, Ankur Taly, Yunhan Jia, Joydeep Ghosh, Ruchir Puri, José MF Moura, and Peter Eckersley. Explainable machine learning in deployment. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 648–657, 2020.

[20] Alexander Binder, Gregoire Montavon, Sebastian Lapuschkin, Klaus-Robert Muller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. In *International Conference on Artificial Neural Networks*, pages 63–71. Springer, 2016.

[21] Benjamin Bowe, Yan Xie, Hong Xian, Tingting Li, and Ziyad Al-Aly. Association between monocyte count and risk of incident ckd and progression to esrd. *Clinical Journal of the American Society of Nephrology*, 12(4):603–613, 2017.

[22] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[23] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[24] Leo Breiman et al. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, 16(3):199–231, 2001.

[25] Javier Castro, Daniel Gómez, and Juan Tejada. Polynomial calculation of the shapley value based on sampling. *Computers & Operations Research*, 36(5):1726–1730, 2009.

[26] Javier Castro, Daniel Gómez, Elisenda Molina, and Juan Tejada. Improving polynomial estimation of the shapley value by stratified random sampling with optimum allocation. *Computers & Operations Research*, 82:180–188, 2017.

[27] Han Soo Chang, Kazuhiro Hongo, and Hiroshi Nakagawa. Adverse effects of limited hypotensive anesthesia on the outcome of patients with subarachnoid hemorrhage. *Journal of neurosurgery*, 92(6):971–975, 2000.

[28] A Charnes, B Golany, M Keane, and J Rousseau. Extremal principle solutions of games in characteristic function form: core, Chebychev and Shapley value generalizations. In *Econometrics of Planning and Efficiency*, pages 123–133. Springer, 1988.

[29] Hugh Chen, Joseph D Janizek, Scott Lundberg, and Su-In Lee. True to the model or true to the data? *arXiv preprint arXiv:2006.16234*, 2020.

[30] Hugh Chen, Scott Lundberg, and Su-In Lee. Explaining models by propagating shapley values of local components. In *Explainable AI in Healthcare and Medicine*, pages 261–270. Springer, 2021.

[31] Hugh Chen, Scott M Lundberg, Gabriel Erion, Jerry H Kim, and Su-In Lee. Forecasting adverse surgical events using self-supervised transfer learning for physiological signals. *NPJ Digital Medicine*, 4(1):1–13, 2021.

[32] Hugh Chen, Ian C. Covert, Scott M. Lundberg, and Su-In Lee. Algorithms to estimate shapley value feature attributions, 2022. URL https://arxiv.org/abs/2207.07605.

[33] Jianbo Chen and Michael Jordan. Ls-tree: Model interpretation when the data are linguistic. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3454–3461, 2020.

[34] Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan. L-shapley and c-shapley: Efficient model interpretation for structured data. *arXiv preprint arXiv:1808.02610*, 2018.

[35] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[36] Yushi Chen, Hanlu Jiang, Chunyang Li, Xiuping Jia, and Pedram Ghamisi. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 54(10):6232–6251, 2016.

[37] Wojtek J Chodzko-Zajko, David N Proctor, Maria A Fiatarone Singh, Christopher T Minson, Claudio R Nigg, George J Salem, and James S Skinner. Exercise and physical activity for older adults. *Medicine & science in sports & exercise*, 41(7):1510–1530, 2009.

[38] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.

[39] Ian Covert and Su-In Lee. Improving kernelshap: Practical shapley value estimation via linear regression. *arXiv preprint arXiv:2012.01536*, 2020.

[40] Ian Covert and Su-In Lee. Improving kernelshap: Practical shapley value estimation using linear regression. In *International Conference on Artificial Intelligence and Statistics*, pages 3457–3465. PMLR, 2021.

[41] Ian Covert, Scott Lundberg, and Su-In Lee. Understanding global feature contributions with additive importance measures. *arXiv preprint arXiv:2004.00668*, 2020.

[42] Ian Covert, Scott Lundberg, and Su-In Lee. Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research*, 22(209):1–90, 2021.

[43] Ian Covert, Chanwoo Kim, and Su-In Lee. Learning to estimate shapley values with vision transformers. *arXiv preprint arXiv:2206.05282*, 2022.

[44] Christine S Cox. *Plan and operation of the NHANES I Epidemiologic Followup Study, 1992*. Number 35. National Ctr for Health Statistics, 1998.

[45] Christine S Cox, Jacob J Feldman, Cordell D Golden, Madelyn A Lane, Jennifer H Madans, Michael E Mussolino, and Sandra T Rothwell. Plan and operation of the nhanes i epidemiologic followup study, 1992. 1997.

[46] Christina Curtis, Sohrab P Shah, Suet-Feung Chin, Gulisa Turashvili, Oscar M Rueda, Mark J Dunning, Doug Speed, Andy G Lynch, Shamith Samarajiwa, Yinyin Yuan, et al. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486(7403):346–352, 2012.

[47] Anupam Datta, Shayak Sen, and Yair Zick. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *2016 IEEE symposium on security and privacy (SP)*, pages 598–617. IEEE, 2016.

[48] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.

[49] Ashley Deeks. The judicial demand for explainable artificial intelligence. *Columbia Law Review*, 119(7):1829–1850, 2019.

[50] Xiaotie Deng and Christos H Papadimitriou. On the complexity of cooperative solution concepts. *Mathematics of operations research*, 19(2):257–266, 1994.

[51] Pam Dixon and Robert Gellman. The scoring of america. In *World Privacy Forum*, 2014.

[52] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[53] Michael Doumpos and Constantin Zopounidis. Model combination for credit risk assessment: A stacked generalization approach. *Annals of Operations Research*, 151(1): 289–306, 2007.

[54] Didier Dreyfuss, Paul Soler, Guy Basset, and Georges Saumon. High inflation pressure pulmonary edema: respective effects of high airway pressure, high tidal volume, and positive end-expiratory pressure. *American Review of Respiratory Disease*, 137(5): 1159–1164, 1988.

[55] Pradeep Dubey, Abraham Neyman, and Robert James Weber. Value theory without efficiency. *Mathematics of Operations Research*, 6(1):122–128, 1981.

[56] Jesse M Ehrenfeld, Luke M Funk, Johan Van Schalkwyk, Alan F Merry, Warren S Sandberg, and Atul Gawande. The incidence of hypoxemia during surgery: evidence from two institutions. *Canadian Journal of Anesthesia/Journal canadien d'anesthésie*, 57(10):888–897, 2010.

[57] Kawin Ethayarajh and Dan Jurafsky. Attention flows are shapley value explanations. *arXiv preprint arXiv:2105.14652*, 2021.

[58] Fatemeh Fahimi, Zhuo Zhang, Wooi Boon Goh, Tih-Shih Lee, Kai Keng Ang, and Cuntai Guan. Inter-subject transfer learning with end-to-end deep convolutional neural network for eeg-based bci. *Journal of neural engineering*, 2018.

[59] Ulrich Faigle and Walter Kern. The shapley value for cooperative games under precedence constraints. *International Journal of Game Theory*, 21(3):249–266, 1992.

[60] Fangfang Fan, Jia Jia, Jianping Li, Yong Huo, and Yan Zhang. White blood cell count predicts the odds of kidney function decline in a chinese community-based population. *BMC nephrology*, 18(1):190, 2017.

[61] Shaheen S Fatima, Michael Wooldridge, and Nicholas R Jennings. A linear approximation method for the shapley value. *Artificial Intelligence*, 172(14):1673–1699, 2008.

[62] Bill Fay. Credit scoring: Fico, vantagescore; other models, 2020. URL https://www.debt.org/credit/report/scoring-models/.

[63] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3429–3437, 2017.

[64] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333, 2015.

[65] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[66] Christopher Frye, Damien de Mijolla, Laurence Cowton, Megan Stanley, and Ilya Feige. Shapley-based explainability on the data manifold. *arXiv preprint arXiv:2006.01272*, 2020.

[67] Christopher Frye, Colin Rowat, and Ilya Feige. Asymmetric shapley values: incorporating causal knowledge into model-agnostic explainability. *Advances in Neural Information Processing Systems*, 33, 2020.

[68] Katsushige Fujimoto, Ivan Kojadinovic, and Jean-Luc Marichal. Axiomatic characterizations of probabilistic and cardinal-probabilistic interaction indices. *Games and Economic Behavior*, 55(1):72–99, 2006.

[69] Robert Geirhos, Jorn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[70] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pages 2242–2251. PMLR, 2019.

[71] Amirata Ghorbani and James Zou. Neuron shapley: Discovering the responsible neurons. *arXiv preprint arXiv:2002.09815*, 2020.

[72] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 513–520, 2011.

[73] Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, 38(3):50–57, 2017.

[74] Daniel Granot, Jeroen Kuipers, and Sunil Chopra. Cost allocation for a tree network with heterogeneous customers. *Mathematics of Operations Research*, 27(4):647–661, 2002.

[75] SPRINT Research Group. A randomized trial of intensive versus standard blood-pressure control. *New England Journal of Medicine*, 373(22):2103–2116, 2015.

[76] Heng Guo and Saul B Gelfand. Classification trees with neural network feature extraction. In *Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 3, pages 183–184. IEEE Computer Society, 1992.

[77] Priyanka Gupta, Pankaj Malhotra, Jyoti Narwariya, Lovekesh Vig, and Gautam Shroff. Transfer learning for clinical time series analysis using deep neural networks. *arXiv preprint arXiv:1904.00655*, 2019.

[78] Stefan Haufe, Frank Meinecke, Kai Görgen, Sven Dähne, John-Dylan Haynes, Benjamin Blankertz, and Felix Bießmann. On the interpretation of weight vectors of linear models in multivariate neuroimaging. *Neuroimage*, 87:96–110, 2014.

[79] Sean P Healey, Warren B Cohen, Zhiqiang Yang, C Kenneth Brewer, Evan B Brooks, Noel Gorelick, Alexander J Hernandez, Chengquan Huang, M Joseph Hughes, Robert E Kennedy, et al. Mapping forest change using stacked generalization: An ensemble approach. *Remote Sensing of Environment*, 204:717–728, 2018.

[80] Tom Heskes, Evi Sijben, Ioan Gabriel Bucur, and Tom Claassen. Causal shapley values: Exploiting causal knowledge to explain individual predictions of complex models. *arXiv preprint arXiv:2011.01625*, 2020.

[81] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. Evaluating feature importance estimates. *arXiv preprint arXiv:1806.10758*, 2018.

[82] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 9737–9748, 2019.

[83] IIF. Iif machine learning recommendations for policymakers. https://www.iif.com/Publications/ID/3574/Machine-Learning-Recommendations-for-Policymakers, 2019.

[84] Ferenc Illés and Péter Kerényi. Estimation of the shapley value by ergodic sampling. *arXiv preprint arXiv:1906.05224*, 2019.

[85] Pari Jahankhani, Vassilis Kodogiannis, and Kenneth Revett. Eeg signal classification using wavelet feature extraction and neural networks. In *IEEE John Vincent Atanasoff 2006 International Symposium on Modern Computing (JVA'06)*, pages 120–124. IEEE, 2006.

[86] Joseph D Janizek, Safiye Celik, and Su-In Lee. Explainable machine learning prediction of synergistic drug combinations for precision cancer medicine. *bioRxiv*, page 331769, 2018.

[87] Dominik Janzing, Lenon Minorics, and Patrick Blobaum. Feature relevance quantification in explainable ai: A causal problem. In *International Conference on Artificial Intelligence and Statistics*, pages 2907–2916. PMLR, 2020.

[88] Sebastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. *arXiv preprint arXiv:1412.2007*, 2014.

[89] Elan Jeremitsky, Laurel Omert, C Michael Dunham, Jack Protetch, and Aurelio Rodriguez. Harbingers of poor outcome the day after severe brain injury: hypothermia, hypoxia, and hypoperfusion. *Journal of Trauma and Acute Care Surgery*, 54(2):312–319, 2003.

[90] Neil Jethani, Mukund Sudarshan, Ian Covert, Su-In Lee, and Rajesh Ranganath. Fastshap: Real-time shapley value estimation. *arXiv e-prints*, pages arXiv–2107, 2021.

[91] Alistair E.w. Johnson, Tom J. Pollard, Lu Shen, Li-Wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, Roger G. Mark, and et al. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3:160035, 2016. doi: 10.1038/sdata.2016.35.

[92] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Zidek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[93] AK Kable, RW Gibberd, and AD Spigelman. Adverse events in surgical patients in australia. *International Journal for Quality in Health Care*, 14(4):269–276, 2002.

[94] Kaggle. The state of ml and data science 2017, 2017. URL https://www.kaggle.com/surveys/2017.

[95] Andrei Kapishnikov, Tolga Bolukbasi, Fernanda Viégas, and Michael Terry. Xrai: Better attributions through regions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4948–4957, 2019.

[96] Adam Karczmarz, Anish Mukherjee, Piotr Sankowski, and Piotr Wygocki. Improved feature importance computations for tree models: Shapley vs. banzhaf. *arXiv preprint arXiv:2108.04126*, 2021.

[97] Warwick D Ngan Kee, Kim S Khaw, Floria F Ng, and Bee B Lee. Prophylactic phenylephrine infusion for preventing hypotension during spinal anesthesia for cesarean delivery. *Anesthesia & Analgesia*, 98(3):815–821, 2004.

[98] Ritva Kiiski, Jukka Takala, Aarno Kari, and J Milic-Emili. Effect of tidal volume on gas exchange and oxygen transport in the adult respiratory distress syndrome. *American Review of Respiratory Disease*, 146:1131–1131, 1992.

[99] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). *arXiv preprint arXiv:1711.11279*, 2017.

[100] Hui Kwon Kim, Goosang Yu, Jinman Park, Seonwoo Min, Sungtae Lee, Sungroh Yoon, and Hyongbum Henry Kim. Predicting the efficiency of prime editing guide rnas in human cells. *Nature Biotechnology*, pages 1–9, 2020.

[101] Hye-Youn Kim, Kyung-Min Lee, So-Hyun Kim, Yeo-Jung Kwon, Young-Jin Chun, and Hyung-Kyoon Choi. Comparative metabolic and lipidomic profiling of human breast cancer cells with different metastatic potentials. *Oncotarget*, 7(41):67111, 2016.

[102] Eric Knight. Ai and machine learning-based credit underwriting and adverse action under the ecoa. *Bus. & Fin. L. Rev.*, 3:236, 2019.

[103] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pages 5338–5348. PMLR, 2020.

[104] Marc D Kohli, Ronald M Summers, and J Raymond Geis. Medical image data and datasets in the era of machine learning—whitepaper from the 2016 c-mimi meeting dataset session. *Journal of digital imaging*, 30(4):392–399, 2017.

[105] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1920–1929, 2019.

[106] PI Korner. Circulatory adaptations in hypoxia. *Physiological reviews*, 39(4):687–730, 1959.

[107] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[108] I Elizabeth Kumar, Suresh Venkatasubramanian, Carlos Scheidegger, and Sorelle Friedler. Problems with shapley-value-based explanations as feature importance measures. In *International Conference on Machine Learning*, pages 5491–5500. PMLR, 2020.

[109] Daniela C Landinez-Lamadrid, Diana G Ramirez-Rios, Dionicio Neira Rodado, Kevin Armando Parra Negrete, and Johana Patricia Combita Nino. Shapley value: its algorithms and application to supply chains. *INGE CUC*, 2017.

[110] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1096, 2019.

[111] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[112] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553): 436–444, 2015.

[113] Su-In Lee, Safiye Celik, Benjamin A Logsdon, Scott M Lundberg, Timothy J Martins, Vivian G Oehler, Elihu H Estey, Chris P Miller, Sylvia Chien, Jin Dai, et al. A machine learning approach to integrate big data for precision medicine in acute myeloid leukemia. *Nature communications*, 9(1):1–13, 2018.

[114] Klas Leino, Shayak Sen, Anupam Datta, Matt Fredrikson, and Linyi Li. Influence-directed explanations for deep convolutional networks. In *2018 IEEE International Test Conference (ITC)*, pages 1–8. IEEE, 2018.

[115] Hong Liang, Xiao Sun, Yunlei Sun, and Yuan Gao. Text feature extraction based on deep learning: a review. *EURASIP Journal on Wireless Communications and Networking*, 2017(1):1–12, 2017.

[116] Andre Lienhart, Yves Auroy, Francoise Pequignot, Dan Benhamou, Josiane Warszawski, Martine Bovet, and Eric Jougla. Survey of anesthesia-related mortality in france. *Anesthesiology: The Journal of the American Society of Anesthesiologists*, 105(6):1087–1097, 2006.

[117] Stan Lipovetsky and Michael Conklin. Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, 17(4):319–330, 2001.

[118] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

[119] Laurent Lonjaret, Olivier Lairez, Vincent Minville, and Thomas Geeraerts. Optimal perioperative management of arterial blood pressure. *Integrated blood pressure control*, 7:49, 2014.

[120] Roberto Lucchetti, Stefano Moretti, Fioravante Patrone, and Paola Radrizzani. The shapley and banzhaf values in microarray games. *Computers & Operations Research*, 37(8):1406–1412, 2010.

[121] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774, 2017.

[122] Scott M Lundberg, Bala Nair, Monica S Vavilala, Mayumi Horibe, Michael J Eisses, Trevor Adams, David E Liston, Daniel King-Wai Low, Shu-Fang Newman, Jerry Kim, et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature biomedical engineering*, 2(10):749–760, 2018.

[123] Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, 2(1):56–67, 2020.

[124] Xinbo Lv, Yi Guan, and Benyang Deng. Transfer learning based clinical concept extraction on data from multiple sources. *Journal of Biomedical Informatics*, 52:55 – 64, 2014. ISSN 1532-0464. doi: https://doi.org/10.1016/j.jbi.2014.05.006. URL http://www.sciencedirect.com/science/article/pii/S1532046414001233. Special Section: Methods in Clinical Research Informatics.

[125] Sumit Majumder, Tapas Mondal, and M Deen. Wearable sensors for remote health monitoring. *Sensors*, 17(1):130, 2017.

[126] Sasan Maleki. *Addressing the computational issues of the Shapley value with applications in the smart grid*. PhD thesis, University of Southampton, 2015.

[127] Pankaj Malhotra, Vishnu TV, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Timenet: Pre-trained deep recurrent neural network for time series classification. *arXiv preprint arXiv:1706.08838*, 2017.

[128] Masayoshi Mase, Art B Owen, and Benjamin Seiler. Explaining black box decisions by shapley cohort refinement. *arXiv preprint arXiv:1911.00467*, 2019.

[129] Sherin M Mathews, Chandra Kambhamettu, and Kenneth E Barner. A novel application of deep learning for single-lead ecg classification. *Computers in biology and medicine*, 99:53–62, 2018.

[130] Yasuko Matsui and Tomomi Matsui. Np-completeness for calculating power indices of weighted majority games. *Theoretical Computer Science*, 263(1-2):305–310, 2001.

[131] Nimrod Megiddo. Computational complexity of the game theory approach to cost allocation for a tree. *Mathematics of Operations Research*, 3(3):189–196, 1978.

[132] Luke Merrick and Ankur Taly. The explanation game: Explaining machine learning models using shapley values. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 17–38. Springer, 2020.

[133] John Merrill, Geoff Ward, Sean Kamkar, Jay Budzik, and Douglas Merrill. Generalized integrated gradients: A practical method for explaining diverse ensembles. *arXiv preprint arXiv:1909.01869*, 2019.

[134] Alexey Miroshnikov, Konstandinos Kotsiopoulos, and Arjun Ravi Kannan. Mutual information-based group explainers with coalition structure for machine learning model explanations. *arXiv preprint arXiv:2102.10878*, 2021.

[135] Rory Mitchell, Joshua Cooper, Eibe Frank, and Geoffrey Holmes. Sampling permutations for shapley value estimation. *arXiv preprint arXiv:2104.12199*, 2021.

[136] Chao-hua Mo, Li Gao, Xiao-fei Zhu, Kang-lai Wei, Jing-jing Zeng, Gang Chen, and Zhen-bo Feng. The clinicopathological significance of ube2c in breast cancer: a study based on immunohistochemistry, microarray and rna-sequencing data. *Cancer Cell International*, 17(1):83, 2017.

[137] Dov Monderer, Dov Samet, et al. Variations on the Shapley value. *Handbook of Game Theory*, 3:2055–2076, 2002.

[138] Matej Moravcik, Martin Schmid, Neil Burch, Viliam Lisy, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337):508–513, 2017.

[139] Jean-François Morcet, Michel Safar, Frédérique Thomas, Louis Guize, and Athanase Benetos. Associations between heart rate and other risk factors in a large french population. *Journal of hypertension*, 17(12):1671–1676, 1999.

[140] Stefano Moretti. Statistical analysis of the shapley value for microarray games. *Computers & operations research*, 37(8):1413–1418, 2010.

[141] Dariush Mozaffarian, Emelia J Benjamin, Alan S Go, Donna K Arnett, Michael J Blaha, Mary Cushman, Sandeep R Das, Sarah de Ferranti, Jean-Pierre Després, Heather J Fullerton, et al. Heart disease and stroke statistics—2016 update: a report from the american heart association. *Circulation*, pages CIR–0000000000000350, 2015.

[142] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019.

[143] Lena Nilsson, Madeleine Borgstedt Risberg, Agneta Montgomery, Rune Sjödahl, Kristina Schildmeijer, and Hans Rutberg. Preventable adverse events in surgical care in sweden: a nationwide review of patient notes. *Medicine*, 95(11), 2016.

[144] Shu Lih Oh, Yuki Hagiwara, U Raghavendra, Rajamanickam Yuvaraj, N Arunkumar, M Murugappan, and U Rajendra Acharya. A deep learning approach for parkinson's

disease diagnosis from eeg signals. *Neural Computing and Applications*, pages 1–7, 2018.

[145] Tomohiko Ohta and Mamoru Fukuda. Ubiquitin and breast cancer. *Oncogene*, 23(11): 2079–2088, 2004.

[146] Ramin Okhrati and Aldo Lipani. A multilinear sampling algorithm to estimate shapley values. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 7992–7999. IEEE, 2021.

[147] Christina Orphanidou. A review of big data applications of physiological signal data. *Biophysical reviews*, 11(1):83–87, 2019.

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

[149] Paolo Palatini. Role of elevated heart rate in the development of cardiovascular disease in hypertension. *Hypertension*, 58(5):745–750, 2011.

[150] Jahan C Penny-Dimri, Christoph Bergmeir, Christopher M Reid, Jenni Williams-Spence, Andrew D Cochrane, and Julian A Smith. Machine learning algorithms for predicting and risk profiling of cardiac surgery-associated acute kidney injury. In *Seminars in Thoracic and Cardiovascular Surgery*. Elsevier, 2020.

[151] Bernard Pereira, Suet-Feung Chin, Oscar M Rueda, Hans-Kristian Moen Vollan, Elena Provenzano, Helen A Bardwell, Michelle Pugh, Linda Jones, Roslin Russell, Stephen-John Sammut, et al. The somatic mutation profiles of 2,433 breast cancers refine their genomic and transcriptomic landscapes. *Nature Communications*, 7(1):1–16, 2016.

[152] Oskar Pfungst. *Clever Hans:(the horse of Mr. Von Osten.) a contribution to experimental animal and human psychology*. Holt, Rinehart and Winston, 1911.

[153] Gregory Plumb, Denali Molitor, and Ameet S Talwalkar. Model agnostic supervised local explanations. In *Advances in Neural Information Processing Systems*, pages 2515–2524, 2018.

[154] Brian Pollard and David B Gibb. Some adverse physiological effects of hypocarbia and methods of maintaining normocarbia during controlled ventilation—a review. *Anaesthesia and intensive care*, 5(2):113–121, 1977.

[155] Arti K Rai. Risk regulation and innovation: the case of rights-encumbered biomedical data silos. *Notre Dame L. Rev.*, 92:1641, 2016.

[156] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning*, pages 759–766. ACM, 2007.

[157] Hariharan Ravishankar, Prasad Sudhakar, Rahul Venkataramani, Sheshadri Thiruvenkadam, Pavan Annangi, Narayanan Babu, and Vivek Vaidya. Understanding the mechanisms of deep transfer learning for medical images. In *Deep Learning and Data Labeling for Medical Applications*, pages 188–196. Springer, 2016.

[158] Jacob Reiter. Developing an interpretable schizophrenia deep learning classifier on fmri and smri using a patient-centered deepshap. In *in 32nd Conference on Neural Information Processing Systems (NeurIPS 2018)(Montreal: NeurIPS)*, pages 1–11, 2020.

[159] Jie Ren, Zhanpeng Zhou, Qirui Chen, and Quanshi Zhang. Learning baseline values for shapley values. *arXiv preprint arXiv:2105.10719*, 2021.

[160] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[161] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI Conference on Artificial Intelligence*, 2018.

[162] Joachim Roski, George W Bo-Linn, and Timothy A Andrews. Creating value in health care through big data: opportunities and policy implications. *Health affairs*, 33(7): 1115–1122, 2014.

[163] Benedek Rozemberczki, Lauren Watson, Péter Bayer, Hao-Tsung Yang, Olivér Kiss, Sebastian Nilsson, and Rik Sarkar. The shapley value in machine learning. *arXiv preprint arXiv:2202.05594*, 2022.

[164] Reuven Y Rubinstein and Dirk P Kroese. *Simulation and the Monte Carlo method*, volume 10. John Wiley & Sons, 2016.

[165] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5): 206–215, 2019.

[166] Luis M Ruiz, Federico Valenciano, and Jose M Zarzuelo. The family of least square values for transferable utility games. *Games and Economic Behavior*, 24(1-2):109–130, 1998.

[167] A. Saabas. treeinterpreter python package. URL https://github.com/andosa/treeinterpreter.

[168] Milad Salem, Shayan Taheri, and Jiann-Shiun Yuan. Ecg arrhythmia classification using transfer learning from 2-dimensional deep cnn features. In *2018 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–4. IEEE, 2018.

[169] Amy J Schmitz. Secret consumer scores and segmentations: Separating haves from have-nots. *Mich. St. L. Rev.*, page 1411, 2014.

[170] Patrick Schwab and Walter Karlen. Cxplain: Causal explanations for model interpretation under uncertainty. In *Advances in Neural Information Processing Systems*, pages 10220–10230, 2019.

[171] Andrew Selbst and Julia Powles. "meaningful information" and the right to explanation. In *Conference on Fairness, Accountability and Transparency*, pages 48–48. PMLR, 2018.

[172] Lloyd Shapley. A value for n-person games. *Contributions to the Theory of Games*, pages 307–317, 1953.

[173] Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016.

[174] Edward H Shortliffe and Martin J Sepúlveda. Clinical decision support in the era of artificial intelligence. *Jama*, 320(21):2199–2200, 2018.

[175] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3145–3153. JMLR. org, 2017.

[176] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[177] Grah Simon and Thouvenot Vincent. A projected stochastic gradient algorithm for estimating shapley value applied in attribute importance. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 97–115. Springer, 2020.

[178] Leon Sixt, Maximilian Granz, and Tim Landgraf. When explanations lie: Why many modified bp attributions fail. In *International Conference on Machine Learning*, pages 9046–9057. PMLR, 2020.

[179] Therese Sørlie, Robert Tibshirani, Joel Parker, Trevor Hastie, James Stephen Marron, Andrew Nobel, Shibing Deng, Hilde Johnsen, Robert Pesich, Stephanie Geisler, et al. Repeated observation of breast tumor subtypes in independent gene expression data sets. *Proceedings of the national academy of sciences*, 100(14):8418–8423, 2003.

[180] Ireneous N Soyiri and Daniel D Reidpath. An overview of health forecasting. *Environmental health and preventive medicine*, 18(1):1, 2013.

[181] Claudia A. Steiner, Zeynal Karaca, Brian J. Moore, Melina C. Imshaug, and Gary Pickens. Surgeries in hospital-based ambulatory surgery and hospital inpatient settings, 2014. *HCUP Statistical Brief*, 2017.

[182] Dave Steinkraus, Ian Buck, and PY Simard. Using gpus for machine learning algorithms. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, pages 1115–1120. IEEE, 2005.

[183] Erik Strumbelj and Igor Kononenko. An efficient explanation of individual classifications using game theory. *The Journal of Machine Learning Research*, 11:1–18, 2010.

[184] Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3): 647–665, 2014.

[185] Erik Štrumbelj, Igor Kononenko, and M Robnik Šikonja. Explaining instance classifications with interactions of subsets of feature values. *Data & Knowledge Engineering*, 68(10):886–904, 2009.

[186] Pascal Sturmfels, Scott Lundberg, and Su-In Lee. Visualizing the impact of feature attribution baselines. *Distill*, 5(1):e22, 2020.

[187] Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43):15545–15550, 2005.

[188] Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. In *International Conference on Machine Learning*, pages 9269–9278. PMLR, 2020.

[189] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

[190] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE Transactions on Medical Imaging*, 35(5):1299–1312, 5 2016. ISSN 0278-0062. doi: 10.1109/TMI.2016.2535302.

[191] Nikola Tarashev, Kostas Tsatsaronis, and Claudio Borio. Risk attribution using the shapley value: Methodology and policy applications. *Review of Finance*, 20(3):1189–1213, 2016.

[192] Nikola A Tarashev, Claudio EV Borio, and Kostas Tsatsaronis. The systemic importance of financial institutions. *BIS Quarterly Review, September*, 2009.

[193] E Taskesen, Aniket Mishra, Sophie van der Sluis, Raffaele Ferrari, Jan H Veldink, MA Van Es, AB Smit, D Posthuma, and Yolande Pijnenburg. Susceptible genes and disease mechanisms identified in frontotemporal dementia and frontotemporal dementia with amyotrophic lateral sclerosis by dna-methylation and gwas. *Scientific Reports*, 7 (1):1–16, 2017.

[194] Kilian Theil and Heiner Stuckenschmidt. Predicting modality in financial dialogue. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 226–234, 2020.

[195] Jeffrey W Tyner, Cristina E Tognon, Daniel Bottomly, Beth Wilmot, Stephen E Kurtz, Samantha L Savage, Nicola Long, Anna Reister Schultz, Elie Traer, Melissa Abel, et al. Functional genomic landscape of acute myeloid leukaemia. *Nature*, 562(7728):526, 2018.

[196] Tjeerd van Campen, Herbert Hamers, Bart Husslage, and Roy Lindelauf. A new approximation method for the shapley value applied to the wtc 9/11 terrorist attack. *Social Network Analysis and Mining*, 8(1):1–12, 2018.

[197] Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge University Press, 2000.

[198] Stephanie M van Rooden, Willem J Heiser, Joost N Kok, Dagmar Verbaan, Jacobus J van Hilten, and Johan Marinus. The identification of parkinson's disease subtypes using cluster analysis: a systematic review. *Movement disorders*, 25(8):969–978, 2010.

[199] Silvia Vanni, Fabio Moda, Marco Zattoni, E Bistaffa, E De Cecco, Marcello Rossi, Giorgio Giaccone, Fabrizio Tagliavini, Stéphane Haïk, Jean-Philippe Deslys, et al. Differential overexpression of serpina3 in human prion diseases. *Scientific Reports*, 7 (1):1–13, 2017.

[200] Joseph Varon and Paul E Marik. Perioperative hypertension management. *Vascular health and risk management*, 4(3):615, 2008.

[201] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michael Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[202] Jiaxuan Wang, Jenna Wiens, and Scott Lundberg. Shapley flow: A graph-based approach to interpreting model predictions. In *International Conference on Artificial Intelligence and Statistics*, pages 721–729. PMLR, 2021.

[203] Rui Wang, Xiaoqian Wang, and David I Inouye. Shapley explanation networks. In *International Conference on Learning Representations*, 2020.

[204] Shuang-Quan Wang, Jie Yang, and Kuo-Chen Chou. Using stacked generalization to predict membrane protein types based on pseudo-amino acid composition. *Journal of Theoretical Biology*, 242(4):941–946, 2006.

[205] Thomas G Weiser, Alex B Haynes, George Molina, Stuart R Lipsitz, Micaela M Esquivel, Tarsicio Uribe-Leitz, Rui Fu, Tej Azad, Tiffany E Chao, William R Berry, et al. Size and distribution of the global volume of surgery in 2012. *Bull World Health Organ*, 94(3):201–209F, 2016.

[206] Daniel S Weld and Gagan Bansal. The challenge of crafting intelligible intelligence. *Communications of the ACM*, 62(6):70–79, 2019.

[207] Tian Wen. An all-payer view of hospital discharge to postacute care, 2013. *HCUP Statistical Brief*, 2016.

[208] Brian Williamson and Jean Feng. Efficient nonparametric statistical inference on population feature importance using shapley values. In *International Conference on Machine Learning*, pages 10282–10291. PMLR, 2020.

[209] David H Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

[210] Yan Xu, Tao Mo, Qiwei Feng, Peilin Zhong, Maode Lai, I Eric, and Chao Chang. Deep learning of feature representation with multiple instance learning for medical image analysis. In *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 1626–1630. IEEE, 2014.

[211] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[212] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

[213] H Peyton Young. Monotonic solutions of cooperative games. *International Journal of Game Theory*, 14(2):65–72, 1985.

[214] Jingbo Yu, Ling Kong, Aihua Zhang, Ying Han, Zhidong Liu, Hui Sun, Liang Liu, and Xijun Wang. High-throughput metabolomics for discovering potential metabolite biomarkers and metabolic mechanism from the appswe/ps1de9 transgenic model of alzheimer's disease. *Journal of Proteome Research*, 16(9):3219–3228, 2017.

[215] Marieke Zegers, Martine C de Bruijne, Bertus de Keizer, Hanneke Merten, Peter P Groenewegen, Gerrit van der Wal, and Cordula Wagner. The incidence, root-causes, and outcomes of adverse events in surgical units: implication for potential prevention strategies. *Patient safety in surgery*, 5(1):13, 2011.

[216] Martin Zinkevich. Rules of machine learning: Best practices for ml engineering, 2017.

[217] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.