Model free Shapley values for high dimensional data

Naofumi Hama\textsuperscript{1}, Masayoshi Mase\textsuperscript{1}, and Art B. Owen\textsuperscript{2}

\textsuperscript{1}Hitachi, Ltd., Research & Development Group
\textsuperscript{2}Department of Statistics, Stanford University

naofumi.hama.hd@hitachi.com, masayoshi.mase.mh@hitachi.com, owen@stanford.edu

November 2022

Abstract

A model-agnostic variable importance method can be used with arbitrary prediction functions. Here we present some model-free methods that do not require access to the prediction function. This is useful when that function is proprietary and not available, or just extremely expensive. It is also useful when studying residuals from a model. The cohort Shapley (CS) method is model-free but has exponential cost in the dimension of the input space. A supervised on-manifold Shapley method from Frye et al. (2020) is also model free but requires as input a second black box model that has to be trained for the Shapley value problem. We introduce an integrated gradient version of cohort Shapley, called IGCS, with cost $O(nd)$.

We show that over the vast majority of the relevant unit cube that the IGCS value function is close to a multilinear function for which IGCS matches CS. We use some area under the curve (AUC) measures to quantify the performance of IGCS. On a problem from high energy physics we verify that IGCS has nearly the same AUCs as CS. We also use it on a problem from computational chemistry in 1024 variables. We see there that IGCS attains much higher AUCs than we get from Monte Carlo sampling. The code is publicly available at https://github.com/cohortshapley/cohortintgrad.

1 Introduction

Quantifying the importance of predictor variables is a first step in explainable AI. As noted by Hooker et al. (2018) and others, there is no ground truth notion of importance. One must instead choose a definition based on such factors as computational feasibility, whether it is defined in terms of ‘off manifold’ data when the underlying features exhibit strong dependence, whether a model must be differentiable, and whether the method can attribute importance to an unused variable which correlates with one that is used, and yet more factors. People can reasonably disagree about whether some of these properties are beneficial or constitute flaws, and we include some discussion of those points below.

Methods from game theory, notably the Shapley value (Shapley 1953) and the Aumann-Shapley value (Aumann and Shapley 1974) are becoming widely used in defining measures of variable importance. When one can reasonably cast the variable importance problem in terms of a game,
then there are persuasive (though still debatable) axioms that lead to a unique definition. The SHapley Additive exPlanations (SHAP) approach of [Lundberg and Lee, 2017] and the Integrated Gradients (IG) of [Sundararajan et al., 2017] are widely cited methods derived from Shapley and Aumann-Shapley values, respectively.

We assume throughout that there are predictor variables $x_i$ with $d$ components (not necessarily real-valued), a response $y_i \in \mathbb{R}$ and a function $f$ where $f(x_i) \in \mathbb{R}$ is the prediction of $y_i$. There are subjects $i = 1, \ldots, n$ and for one or more target subjects $t$, we want to quantify how important $x_{tj}$ is for $f(x_t)$.

One important distinction between methods is whether they are model-based, model-agnostic, or model-free. For example, TreeSHAP [Lundberg et al., 2020] can be computed very efficiently but only for models that have a tree structure, so it is model-based. By contrast, techniques such as the baseline Shapley method in [Sundararajan and Najmi, 2020] can be used on any black box model $f(\cdot)$, and hence are called model-agnostic. In this paper we consider model-free variable importance methods that use only the observed values $(x_i, f(x_i))$ or even $(x_i, y_i)$ for $i = 1, \ldots, n$ to quantify the importance of components $x_{tj}$ on the prediction $f(x_t)$ or the response $y_t$.

The cohort Shapley (CS) method of [Mase et al., 2019] is model-free. However, exact computation of the CS values has a cost that is exponential in the number $d$ of variables within $x_i$. Integrated gradient methods have been proposed for Shapley value by [Sundararajan et al., 2017]. These are versions of the Aumann-Shapley value. In this paper we construct an integrated gradients approximation to CS, which we call IGCS. IGCS has a cost that is $O(nd)$ for $n$ observations.

The supervised on-manifold learning of [Frye et al., 2020] is also model-free. It is a framework for constructing model-free methods, rather than a model-free method itself, as we explain later.

There are several reasons to select a model-free method. [Mase et al., 2021] study the COMPAS model predicting recidivism, from [Angwin et al., 2016]. In this case, only the predictions themselves are available because the underlying prediction function is proprietary and not available to researchers. As a result, methods that require access to the function $f(\cdot)$ cannot be used but CS is model-free and hence could be used to address some algorithmic fairness issues. A second reason to use model-free methods is to study variable importance for given data $(x_i, y_i)$ without reference to any model. A variable that is important for predicting $y_i$ is then one that we reasonably expect to play a role in a good prediction model for $y$. A third motivation for model-free methods is to learn what variables are important for residuals $y_i - f(x_i)$ or the absolute values or squares of those residuals. Even when we have access to $f(\cdot)$ we still do not have a way to study residuals at points $x$ where we do not have the corresponding $y$ value. If a variable is important for residuals, that provides a hint that the model has some flaw that can be improved. In the given data framework it is possible that two or more observations have identical $x$ but different $y$. Both CS and IGCS can handle this setting. That is, in the list of $(x_i, y_i)$ pairs it is not necessary for $y_i$ to be a function of $x_i$.

While most IG methods require differentiability of the prediction function $f$, IGCS does not. It is based on weighted means of observed function values and the necessary gradient is defined in terms of a differentiable weight function.

The remainder of this paper is organized as following. Section 2 cites some related works in the explainable AI (XAI) literature. Section 3 introduces some notation, gives a formal presentation of our IGCS proposal and defines the area under the curve (AUC) and area between the curves (ABC) measures we use to judge the quality of variable importance measures. We also introduce what to our knowledge is a more general setting where integrated gradients match Shapley value beyond the settings from [Owen, 1972] or [Sundararajan et al., 2017]. Section 4 shows via Taylor expansion
that the value function underlying IGCS is very nearly one where IGCS matches CS over the vast majority of its input space provided that the dimension \( d \) satisfies \( d \gg \log(n) \) for \( n \) data points. This finding is obtained under very general conditions on the data, and does not need assumptions strongly tuned to the model form that the usual integrated gradients method does. Section 5 conducts experiments to confirm the feasibility of IGCS. In a 16-dimensional problem from high energy physics it performs almost as well as CS. In a 1024-dimensional problem from computational chemistry it performs much better than a Monte Carlo strategy and the results we inspected made sense in terms of basic chemistry. There are some conclusions in Section 6. Appendix A reports some details of our experimental setup and also derives a model-free version of GKW.

2 Related Works

It is very difficult to quantify the importance of a variable to some output. Beyond the computational issues are problems of defining what constitutes a good measure of importance. This can be based on attaining desired properties or avoiding undesirable ones. In this paper, we work in some game theoretic paradigms. Table 1 shows a grid of properties of game theoretic methods. There is not space to describe all of the methods. For a recent survey of the area see Chen et al. (2022). We say a bit more about some methods below, when we introduce our proposed method.

A method can be ‘local’, explaining the prediction \( f(x_t) \) for observation \( t \) or ‘global’, explaining the value of variable \( j \) in getting an accurate model for all observations \( i = 1, \ldots, n \). We focus on local methods. A method can be based on fixing some components of \( x_t \) while intervening to vary the others or it can be based on fixing the first set of components and summarizing the effects of sampling the others, by for example, a conditional mean. Such methods are called conditional. We prefer conditional methods but include some interventional ones in our comparisons. Whether conditional or interventional, a method can be ‘on-manifold’ attempting to only use \( f \) at realistic inputs \( x_t \), such as values scattered around a manifold within the input space, or it can be ‘off-manifold’, computing with arbitrary combinations of input components.

One debate that arises in variable importance is whether any importance should ever be assigned to a variable that the model function \( f \) does not even use. Generally that happens in conditional methods but not interventional ones. The choice is presented as a catch-22 in Kumar et al. (2020). It can seem absurd to give importance to an unused variable while at the same time, it is necessary to have that possibility in algorithmic fairness contexts. Another tradeoff is that off-manifold methods are reasonable for independent inputs but involve problematic input combinations unlike any seen in past or future data otherwise. Both Hooker and Mentch (2019) and Mase et al. (2019) emphasize this point. On the other hand, on-manifold methods are more challenging to compute. There is even a debate over whether models should be explained at all, instead of just using models that are themselves interpretable (Rudin, 2019).

The supervised on-manifold method of Frye et al. (2020), labeled SS in Table 1, is a framework for constructing model-free Shapley values. They train a neural network to predict \( f(x_t) \) given predictors \( \tilde{x}_t(u) \) where \( u \) is a set of predictor indices. Here \( \tilde{x}_t(u) \) is a vector formed by replacing \( x_{ij} \) by an NA value (not available) for all predictors \( j \in u \). In the notation we introduce in Section 3, \( \tilde{x}_t(u) = \tilde{x}_0, u \cdot x_{1, -u} \) where \( x_0 \) is a tuple of \( d \) NA values. Their NA value was \(-1\) which never appears in the data sets they study. They train with a loss function based on accurately predicting \( f(x_t) \) given, uniformly distributed data points \( i \), a Shapley-relevant distribution on \( u \) and a distribution for \( x_{ij} \) when \( j \not\in u \). There are many choices for the neural network to use, the choice could be customized to the data set at hand, and their framework could be used with other algorithms other
Table 1: Summary of properties of variable importance methods: cohort Shapley (Mase et al., 2019), baseline Shapley (Sundararajan and Najmi, 2020; Chen et al., 2022), TreeSHAP (Lundberg et al., 2020), KernelSHAP (Lundberg and Lee, 2017), integrated gradients (Sundararajan et al., 2017), On-manifold (Frye et al., 2020), surrogate SHAP (Frye et al., 2020) and Conditional Kernel SHAP (Aas et al., 2021), empirical Gaussian Kernel Weight (Appendix A.3) and our integrated gradient cohort Shapley proposal. Properties from top to bottom: uses only observed predictor combinations, uses no independence assumption on inputs, can attribute importance to unused variables, can use any $f$, does not need $f$, cost of exact computation is not exponential in $d$, computes a Shapley value instead of an approximation, does not use a second black box to explain $f$, and requires no user input.

The empirical GKW method is our adaptation of the Gaussian kernel weight method of Aas et al. (2021). See Appendix A.3. The original GKW is model-agnostic but not model-free.

The cost of computing Shapley values exactly is exponential in the number of features because all $2^d$ subsets of $d$ features must be considered. An approximation by integrated gradients, can avoid this exponential cost. For special functions, such as multilinear interpolations, it is known that integrated gradients reproduce the Shapley values exactly. See Owen (1972) and for uses in XAI, see Sundararajan and Najmi (2020).

The statistics literature has several measures of variable importance that can be defined in terms of the joint distribution of random variables without requiring a parametric form for any predictions, and they are in that sense model-free. These include the maximal correlation of Rényi (1959), the energy statistics of Székely and Rizzo (2013), the conditional dependence measures of Azadkia and Chatterjee (2021), the floodgate method of Zhang and Janson (2020). These are commonly defined through conditional variances and do not give additive local explanations of $f(x_t)$ for a given subject $t$, so they don’t fit into our comparisons.

3 Notation, background and the method

Variable importance measures study a mapping $f : \mathcal{X} \to \mathbb{R}$ that makes a prediction based on input data $x \in \mathcal{X} = \prod_{j=1}^{d} \mathcal{X}_j$. The points $x \in \mathcal{X}$ are written $x = (x_1, x_2, \ldots, x_d)$. We have $n$ observations $x_1, \ldots, x_n$ and the goal is to quantify importance of the $d$ components of $x_t$ to $f(x_t)$ (or $y_t$) where $t$
is the ‘target’ observation. Because we are working in a model-free setting we only need to compute
with the values \((x_i, f(x_i))\) for \(i = 1, \ldots, n\). We can also relax the assumption that \(f\) is a mapping.
For instance, we can replace values \(f(x_i)\) by values \(y_i\) where \(x_i = x_i'\) need not imply \(y_i = y_i'\).

We use \([d]\) for the set \(\{1, \ldots, d\}\). For \(u \subseteq [d]\) we write \(|u|\) for the cardinality of \(u\) and \(-u\) for \([d] \setminus u\).
For singletons we write \(-j\) in subscripts instead of \(-\{j\}\). The subvector \(x_u\) has all
components \(x_j\) for \(j \in u\) and no others. We write \(x_{tu}\) for the corresponding subvector of \(x_t\).
We will refer to hybrid points \(x_u; z_{-u} \in \mathcal{X}\). If \(y = x_u; z_{-u}\) then \(y_{-j} = x_j\) for \(j \in u\) and \(y_{-j} = z_j\) for \(j \not\in u\).

Because we are attributing importance of \(f(x_t)\) to the components \(x_{tj}\) for \(j = 1, \ldots, d\) the
problem is one of ‘local importance’. As noted above, this is distinct from ‘global importance’
problems of identifying which variables must be included in the model in order to get accurate
predictions.

We will use a weight space \([0, 1]^d\). In this space we write \(1 = (1, 1, \ldots, 1)\) and \(0 = (0, 0, \ldots, 0)\).
Those two points belong to \((0, 1)^d\) which we describe as the set of ‘corners’ of \([0, 1]^d\). A general
corner is of the form \(1_u; 0_{-u}\) for \(u \subseteq [d]\).

### 3.1 Shapley value and CS

We begin by describing the Shapley value that underlies the game theoretic methods. In cooperative
game theory, we suppose that a team of \(d\) players has created some value that we must attribute
to them as individuals. We assume that we know the value \(\nu(u)\) that would have been created by
any subteam \(u \subseteq [d]\) of those participants and suppose for now that \(\nu(\emptyset) = 0\). If \(j \not\in u\) then the
incremental value from adding player \(j\) to team \(u\) is

\[
\nu(j | u) = \nu(u \cup \{j\}) - \nu(u).
\]

The Shapley value for player \(j\) is the following weighted sum of its incremental values:

\[
\phi_j = \frac{1}{d} \sum_{u \subseteq -j} \binom{d - 1}{|u|}^{-1} \nu(j | u). \tag{1}
\]

We can also get this value by building a team from \(\emptyset\) to \([d]\) in one of \(d!\) orders, adding one player at a
time, and taking \(\phi_j\) to be the average incremental value from the addition of player \(j\) over those
\(d!\) orders.

Shapley derived the value \(\phi_j\) in [1] as the unique solution compatible with four axiomatic
criteria:

1. **Efficiency** \(\sum_j \phi_j = \nu([d])\).
2. **Dummy** If \(\nu(j | u) = 0\) for all \(u \subseteq -j\) then \(\phi_j = 0\).
3. **Symmetry** If \(\nu(j | u) = \nu(j' | u)\) when \(u \subseteq -\{j, j'\}\) then \(\phi_j = \phi_{j'}\).
4. **Additivity** If \(\nu\) and \(\nu'\) have values \(\phi_j\) and \(\phi_j'\) then \(\nu + \nu'\) has values \(\phi_j + \phi_j'\).

We find it convenient to drop the assumption that \(\nu(\emptyset) = 0\). Then the efficiency condition
generalizes to \(\sum_j=1 \phi_j = \nu([d]) - \nu(\emptyset)\). That is, we are still explaining \(\nu([d]) - \nu(\emptyset)\) without being
concerned over whether \(\nu(\emptyset) = 0\).

Using Shapley value we can derive importances for all \(d\) variables by analogy where \(\nu(u)\) is a
measure of how well the variables in \(x_u\) explain \(f(x)\). For instance the LMG measure of [Lindeman et al. (1980)] is the Shapley value when \(\nu(u)\) is the \(R^2\) value in a linear model relating \(y\) to \(x_u\). This
is a global measure of variable importance for all observations not just the target \(x_t\).
An easily described local measure is baseline Shapley where \( \nu(u) = f(x_{t,u}; x_{b,-u}) \). Here \( x_t \) is the target point and we seek to explain the difference between \( f(x_t) \) and \( f(x_b) \) where \( x_b \in \mathcal{X} \) is some baseline or default point. It could be a real point or the average of all \( n \) points. Baseline Shapley is considered an ‘interventional’ method because it is implemented by intervening to change some of the components of \( x_b \) to those of \( x_t \).

Another measure is conditional expectation Shapley where \( \nu(u) = \mathbb{E}(f(x_t \mid x_{tu})) \). More precisely, conditional expectation Shapley is a family of measures because multiple choices for the conditional distribution of \( x_t \) given \( x_{tu} \) have been considered. Random Baseline Shapley methods (Sundararajan and Najmi 2020; Lundberg and Lee 2017) assume that the components of \( x_{t,-u} \) are drawn independently from the marginal distributions of \( x \). Aas et al. (2021) estimate the conditional distribution using kernel methods.

One difficulty with baseline Shapley and other interventional methods is that they can use extremely unlikely or even physically impossible variable combinations. If the point \( x_{t,u}; x_{b,-u} \) is completely impossible then the value \( f(x_{t,u}; x_{b,-u}) \) may be meaningless and hence not fit for use. The cohort Shapley method of Mase et al. (2019) was constructed to avoid using impossible combinations. It does so by only using actually observed data. It begins with user-specified notions of whether the value \( x_{ij} \) is similar to \( x_{ij} \). For binary features or those with a small number of levels it is natural to take similarity to mean that \( x_{ij} = x_{ij} \). For continuously distributed variables, one can take similarity to mean \( |x_{ij} - x_{ij}| \leq \delta_j \) or one can discretize \( \mathcal{X}_j \) into a modest number of buckets. It must always be true that \( x_{ij} \) is similar to itself. If all \( d \) variables have similarity defined in terms of equality, perhaps after discretization, then CS becomes conditional expectation Shapley based on the empirical distribution of \( x_t \). The requirement to define similarity is a burden. However, alternatives that use a black box to estimate conditional expectations are less transparent in that the effective definition of similarity is obscured.

We let \( S_j(x_i) \) be 1 if \( x_{ij} \) is similar to \( x_{ij} \) and 0 otherwise. For \( u \subseteq [d] \), we let \( S_u(x_i) = \prod_{j \in u} S_j(x_i) \) with \( S_{\emptyset} = 1 \) by convention. The value function for CS is

\[
\nu(u) = \nu(u; t) = \frac{\sum_{i=1}^{n} f(x_i) S_u(x_i)}{\sum_{i=1}^{n} S_u(x_i)}.
\]

It is the average value of \( f(x_i) \) over the cohort \( C_u := \{ i \in [d] \mid S_u(x_i) = 1 \} \) of observations similar to the target \( x_t \) for all \( j \in u \). For \( x_t \in C_u \) we might or might not have \( S_j(x_i) = 1 \) for some \( j \not\in u \). The cohort mean \( \nu(u) \) is always well defined because \( C_u \) is never empty, as \( t \in C_u \). With this definition the Shapley value explains

\[
\frac{1}{C_{[d]} \setminus \{t\}} \sum_{x \in C_{[d]} \setminus \{t\}} f(x_t) - \frac{1}{n} \sum_{i=1}^{n} f(x_i).
\]  \[\text{(2)}\]

The mean on the left of (2) is the average of \( f(x_i) \) over observations \( i \) in the ‘fully refined cohort’ \( C_{[d]} \) of observations similar to \( x_t \) on all \( d \) variables. Very commonly \( C_{[d]} = \{t\} \). Then CS attributes the difference \( f(x_t) - (1/n) \sum_{i=1}^{n} f(x_i) \) to the \( d \) individual input variables through the Shapley values \( \phi_j \) that correspond to \( u \).

The set of Shapley values \( \phi_1, \ldots, \phi_d \) depend on \( d2^{d-1} \) incremental values. For special constructions of \( f \) there are faster algorithms, with TreeSHAP (Lundberg et al. 2020) being a well known special case, but for model-agnostic Shapley values, we cannot avoid the exponential cost.
3.2 Integrated gradients and IGCS

If $f$ is differentiable and each $X_i$ is an interval subset of $\mathbb{R}$, with finite positive length, then the integrated gradients method from [Sundararajan et al. (2017)] is a computationally attractive alternative to Shapley value methods. We assume that the variables $x$ have been rescaled to belong to the unit cube $[0,1]^d$. With IG we assign the values

$$\psi_1, \ldots, \psi_d = \int_0^1 \nabla f(z) \, dz \quad \text{where} \quad \nabla f(z) = (\frac{\partial}{\partial z_1} f(z), \ldots, \frac{\partial}{\partial z_d} f(z))$$

in order to explain $f(1) - f(0)$. The corresponding Shapley value that we would like is the one with value function $\nu(u) = f(\mathbf{1}_u, \mathbf{0}_{-u})$. That would be the baseline Shapley value if our mapping was such that $x_i$ corresponded to $z = 1$ and $x_b$ corresponded to $z = 0$. We readily find by the fundamental theorem of calculus that $f(1) - f(0) = \sum_{j=1}^d \psi_j$, so IG satisfies the efficiency axiom.

We want a CS analogue to IG. To get an approximation to CS that scales to high dimensional settings, we introduce a soft similarity function

$$s_z(x_i) = s_z(x_i; t) = \prod_{j=1}^d (1 + z_j(S_j(x_i) - 1))$$

for $z \in [0,1]^d$. Here $1 + z_j(S_j(x_i) - 1)$ is a soft version of $S_j(x_i)$ that linearly interpolates from 1 to $S_j(x_i) \in [0,1]$ as $z$ increases from 0 to 1. Soft similarity satisfies $s_{1_u,0_{-u}}(x_i) = S_u(x_i)$. We can now extend the CS value $\nu$ from $u \subseteq [d]$ to $z \in [0,1]^d$ via

$$\nu(z) = \frac{\sum_{i=1}^n f(x_i) s_z(x_i)}{\sum_{i=1}^n s_z(x_i)}.$$  (3)

We interpret the numerator of $\nu(z)$ as a soft total and the denominator as a soft cardinality. Note that while the components $z_j$ are real values in $[0,1]$, the data features $x_j \in X$ do not have to be real-valued. As a result, the soft value function $\nu$ in (3) can be defined for very general features. The quantity explained is $\nu(1) - \nu(0)$ which reduces to the fully refined cohort mean minus the sample mean as in CS. The cohort Shapley value function for the set $u$ is now the soft cohort value $\nu(1_u,0_{-u})$.

The integrated gradient cohort Shapley (IGCS) values are given by

$$\psi_1, \ldots, \psi_d = \int_0^1 \nabla \nu(z) \, dz \quad \text{where} \quad \nabla \nu(z) = (\frac{\partial}{\partial z_1} \nu(z), \ldots, \frac{\partial}{\partial z_d} \nu(z)).$$

The partial derivatives we need depend on

$$s_z^{(k)}(x_i) \equiv \frac{\partial}{\partial z_k} s_z(x_i) = (S_k(x_i) - 1) \prod_{j \neq k} (1 + z_j(S_j(x_i) - 1)).$$

Now

$$\frac{\partial}{\partial z_k} \nu(z) = \frac{\sum_{i=1}^n f(x_i) s_z^{(k)}(x_i)(\sum_{i=1}^n s_z(x_i)) - (\sum_{i=1}^n f(x_i) s_z(x_i))(\sum_{i=1}^n s_z^{(k)}(x_i))}{(\sum_{i=1}^n s_z(x_i))^2}.$$
This partial derivative can be computed at cost $O(nd)$. Integrating it with a quadrature rule on $R$ nodes then costs $O(nRd)$. Now suppose that we evaluate the gradient at a point $z\mathbf{1}$. We get

$$s_{z\mathbf{1}}(x_i) = \prod_{j=1}^{d}(1 + z(S_j(x_i) - 1)), \quad \text{and}$$

$$s_{z\mathbf{1}}^{(k)}(x_i) = (S_k(x_i) - 1) \prod_{j \in -k} (1 + z(S_j(x_i) - 1)).$$

The partial derivative is then a rational function in $z$ with a numerator degree of $2d - 1$ and a denominator degree of $2d$. We will estimate its integral via an equally weighted average over equispaced points $z_r \in [0, 1]$ for $r = 1, \ldots, R$.

When we devise a differentiable function on $[0, 1]^d$ interpolating some desired values on the corners $\{0, 1\}^d$ we might ask when the integrated gradients values on $[0, 1]^d$ match the Shapley values. It is known from Sun and Sundararajan (2017) that this match happens for functions $f : [0, 1]^d \to \mathbb{R}$ that are multilinear such as $\prod_{j \in u} z_j$ or sums of multilinear functions with a denominator degree of 2.

We note here that the agreement generalizes beyond those above mentioned cases. We have not seen this more general agreement in the literature. While it may be previously known in game theory, it does not appear in Owen (1972) and was not known in the XAI literature as of Sundararajan et al. (2017). Our argument uses each Shapley axiom once.

**Theorem 1.** Let $h : [0, 1] \to \mathbb{R}$ be a differentiable function and for nonempty $u \subseteq [d]$ let $f(z) = \prod_{j \in u} h(z_j)$. Then integrated gradients $\psi_j$ for $f(z)$ on $[0, 1]^d$ match the Shapley values $\phi_j$ for $\nu(u) = f(1_u, \mathbf{0}_{-u})$.

**Proof.** From the dummy axiom, $\phi_j$ is zero for $j \not\in u$. From the symmetry axiom $\phi_j$ for $j \in u$ are all equal. From the efficiency axiom they must sum to $h(1)|u| - h(0)|u|$. Therefore $\phi_j = 1_{j \in u}(h(1)|u| - h(0)|u|)/|u|$. Similarly for $j \in u$

$$\psi_j = \int_0^1 h'(z)h(z)^{|u| - 1}dz = \left[\frac{h(z)^{|u|}}{|u|}\right]_0^1 = \frac{h(1)|u| - h(0)|u|}{|u|}$$

while for $j \not\in u$ the $j$’th component of $\nabla \nu$ is zero. □

**Corollary 1.** Let

$$f(z) = \sum_{\emptyset \neq u \subseteq [d]} L_u \sum_{\ell=1}^{L_u} \prod_{j \in u} h_{u, \ell}(z_j)$$

for arbitrary differentiable functions $h_{u, \ell}$ on $[0, 1]$. Then the integrated gradients $\psi_j$ for $f(z)$ on $[0, 1]^d$ match the Shapley values $\phi_j$ for $\nu(u) = f(1_u, \mathbf{0}_{-u})$.

**Proof.** The result follows from Theorem 1 and the additivity axiom of Shapley value. □

We can of course add a constant term ($u = \emptyset$) to the function $f(z)$ above. For $|u| = 1$ we do not need $L_u > 1$, but for $|u| > 1$ allowing $L_u > 1$ provides a generalization.
3.3 Deletion and insertion measures

As noted earlier, there is no ground truth for variable importance. Instead there are multiple definitions of what makes a variable important and choosing a definition involves some tradeoffs. It is however possible to compute the value of a proxy measure derived from an area under the curve (AUC) quantity defined by Petsiuk et al. (2018). Our presentation of that measure is based on the analysis of AUC in Hama et al. (2022).

The proxy measure is about the quality with which the variables' importances can be ranked. For an interventional method like baseline Shapley the quality of a ranking can be measured as follows. We sort the variables from those with the largest $\phi_j$ to those with the smallest $\phi_j$. For $k = 0, 1, \ldots, d$ let $\tilde{x}_k$ be a hybrid point with $\tilde{x}_{kj} = x_{tj}$ if variable $j$ is among the $k$ variables with the largest $\phi_j$ and let $\tilde{x}_{kj} = x_{bj}$ otherwise. Then let $\tilde{y}_k = f(\tilde{x}_k)$ and consider the piecewise linear curve going through the points $(k, \tilde{y}_k)$ over the interval $[0, d]$. The AUC criterion is the (signed) area under this curve. If the variables have been well ranked then the AUC of the insertion curve will be large. Hama et al. (2022) describe an area between the curves (ABC) that subtracts the area under a straight line connecting $(0, \tilde{y}_0)$ to $(d, \tilde{y}_k)$. If one method has a better AUC than another it will also have a better ABC. That paper argues that ABC is better suited to regression problems than AUC which was devised for classification.

There is a similar deletion curve found by changing the variables in the reverse order. Hama et al. (2022) define an ABC for deletion as the signed area under the straight line minus the area under the deletion curve.

Ranking is not the same as estimating the $\phi_j$. Hama et al. (2022) show an example where ranking by the true Shapley values can give a lesser ABC than some other ranking. They also show that for certain simple models, such as logistic regression and naive Bayes, that ranking by Shapley value does maximize the ABC. Since those models often perform well, we expect that more sophisticated models will often make similar predictions and then ABC will be a useful scorecard.

On our data examples it is impossible to compute the true Shapley values so we use ABC measures as a guide to compare CS to IGCS and some other methods.

Just as there is more than one way to define variable importance, there is more than one way to define an ABC measure to rank variable importance measures. Instead of the interventional approach above one can replace $\tilde{y}_k$ by the average value of $f(x_t)$ over the cohort of points that are similar to $x_t$ for the $k$ variables with the greatest value of $\phi_j$. This provides a conditional ABC measure as an alternative to an interventional ABC measure.

We expect on intuitive grounds that a method which attempts to compute interventional Shapley values will attain a better interventional ABC value than one that attempts to compute conditional Shapley values and vice versa. While we do not prove that this must happen, we will see it happen in our numerical examples. We have a strong preference to avoid interventional measures as they can require evaluating $f$ at some wildly unrealistic input values, but other researchers might accept those measures.

4 CS versus IGCS for large $d$

In the CERN example of Section 5 with $d = 16$, we will see that cohort Shapley and integrated gradient cohort Shapley attained very nearly the same ABC values. We know for sure that IGCS cannot always match CS because CS requires exponential computation and IGCS does not. In this section we explain why CS and IGCS can be expected to be very close to each other in high
dimensional settings where CS cannot be computed, such as a 1024-dimensional example from computational chemistry, also in Section 5.

While both the numerator and denominator of $\nu(z)$ are multilinear functions where the integrated gradient recovers the Shapley values, this is not true of the ratio itself. We use a Taylor expansion of the soft cohort mean around the point $1 \in [0,1]^d$ to explain how the ratio is very nearly multilinear over most, though not all, of $[0,1]^d$.

It helps to introduce dissimilarity sets

$$J_i = \{ j \in [d] \mid S_j(x_i) = 0 \}.$$ 

Then the soft similarity for observation $i$ is

$$s_z(x_i) = \prod_{j \in J_i} (1 - z_j).$$

Now we make a formal Taylor series expansion

$$\nu(z) = \left( f(x_t) + \sum_{i \neq t} f(x_i) \prod_{j \in J_i} (1 - z_j) \right) / \left( 1 + \sum_{i \neq t} s_z(x_i) \right)$$

$$= \left( f(x_t) + \sum_{i \neq t} f(x_i) \prod_{j \in J_i} (1 - z_j) \right) \left( 1 + \sum_{r=1}^{\infty} \left( - \sum_{i \neq t} s_z(x_i) \right)^r \right).$$

This formal expansion fails to converge for some values of $z$. We next give conditions under which it converges for the vast majority of $[0,1]^d$. Those conditions keep the soft cardinality between 1 and $1 + \epsilon$ over the vast majority of $[0,1]^d$. The lower bound is 1 because the target point $x_t$ is always counted.

We assume that the fraction of variables $j \in [d]$ for which $x_i$ is dissimilar to $x_t$ belongs to a sub-interval of $(0,1]$ as follows:

$$ad \leq |J_i| \leq Ad$$

for constants $0 < a \leq A \leq 1$.

**Theorem 2.** Let there be $n$ observations $x_i$ for $i \neq t$ and suppose that equation 5 holds for the target point $x_t$ and similarity functions $S_j(x_i)$. For $0 < \epsilon < 1$ let

$$H_\epsilon = \left\{ z \in [0,1]^d \mid \sum_{i \neq t, j \in J_i} (1 - z_j) \geq \epsilon \right\}.$$ 

Then for $z \sim U[0,1]^d$,

$$\Pr(z \in H_\epsilon) \leq \frac{n^2 \epsilon}{\epsilon} \exp\left(-\frac{|ad|}{4}\right).$$

**Proof.** We will use the fact that minus twice the sum of the logs of $k$ independent $U(0,1)$ random
variables has the \( \chi^2_{(k)} \) distribution. We get
\[
\Pr(z \in H_\epsilon) \leq \Pr \left( \max_{i \neq t} \prod_{j \in J_i} (1 - z_j) \geq \frac{\epsilon}{n} \right) \\
\leq n \Pr \left( \prod_{j=1}^{|ad|} (1 - z_j) \geq \frac{\epsilon}{n} \right) \\
= n \Pr \left( -2 \sum_{j=1}^{|ad|} \log(1 - z_j) \leq -2 \log(\epsilon/n) \right) \\
= n \Pr(\chi^2_{(2|ad|)} \leq 2 \log(n/\epsilon)).
\]
Now equation (4.4) of Laurent and Massart (2000) shows that for any integer \( \nu \geq 1 \) and any \( x > 0 \)
\[
\Pr(\chi^2_{(\nu)} \leq \nu - 2\sqrt{\nu x}) \leq e^{-x}.
\]
Taking \( \nu = 2|ad| \) and \( x = (\nu - 2\log(n/\epsilon))^2/(4\nu) \) we get
\[
\Pr(z \in H_\epsilon) \leq n \exp \left( -\frac{(\nu - 2\log(n/\epsilon))^2}{4\nu} \right) \\
\leq n \exp \left( -\frac{|ad|}{4} + \log \left( \frac{n}{\epsilon} \right) \right).
\]
Moving \( \log(n/\epsilon) \) out of the exponent completes the proof.

As a result, we conclude that in a setting with \( d \gg \log(n) \), the Taylor expansion (4) is convergent over all but a trivially small part of the unit cube. The theorem above is for \( n \) points in addition to the target point \( x_t \). For \( n - 1 \) such points the bound is the somewhat less elegant \( (n - 1)^2 \exp(-|ad|/4)/\epsilon \).

Remark 1. The bound in Theorem 2 does not involve the constant \( A \). This means that observations where \( x_i \) is dissimilar to \( x_t \) for most or even all features are not detrimental to the convergence of the Taylor series.

Remark 2. At the other extreme, it is possible that some point \( x_i \) is essentially a duplicate of \( x_t \) and is then similar to \( x_t \) for all \( d \) predictor variables. We write \( i \sim t \) for that case and \( i \not\sim t \) otherwise. Suppose that there are \( n_1 \geq 1 \) points identical to \( x_t \) (including \( x_t \) itself) and also \( n_2 \) other points satisfying equation (5). Then the soft cohort mean is
\[
\left( \sum_{i: i \sim t} f(x_i) + \sum_{i: i \not\sim t} f(x_i) \prod_{j \in J_i} (1 - z_j) \right) / \left( n_1 + \sum_{i: i \not\sim t} \prod_{j \in J_i} (1 - z_j) \right) \\
= \left( \bar{f} + \frac{1}{n_1} \sum_{i: i \sim t} f(x_i) \prod_{j \in J_i} (1 - z_j) \right) / \left( 1 + \frac{1}{n_1} \sum_{i: i \not\sim t} \prod_{j \in J_i} (1 - z_j) \right)
\]
where \( \bar{f} = \sum_{i: i \sim t} f(x_i)/n_1 \). The Taylor expansion in equation (4) is convergent over the vast majority of \([0, 1]^d\) for this soft cohort mean too, by the same argument we used in Theorem 2. This holds whether \( n_1 = O(1) \) or whether \( n_1 \) grows proportionally to \( n \). We just need \( d \gg \log(n_2) \).
Now, we turn to the corners of the cube used in the definition of cohort Shapley and show that the Taylor expansion is convergent for the vast majority of them. Clearly \( 0 \in H_z \) and \( 1 \not\in H_z \). We let \( z = 1_u:0_v, u, v \in \{0,1\}^d \).

**Theorem 3.** Let there be \( n \) observations including \( x_i \) and suppose that equation (5) holds for the target point \( x_i \) and similarity functions \( S_j(x_i) \). Then the Taylor expansion (4) is convergent for at least

\[
2^d \left( 1 - \frac{n}{2da} \right)
\]

of the points in \( \{0,1\}^d \).

**Proof.** For a corner point \( z \in \{0,1\} \) to be in \( H_z \) of equation (4) there must be at least one observation \( i \) with \( \prod_{j \in J_i} (1 - z_j) \neq 0 \). That is \( u \subseteq J_i' \). There are at most \( 2^{d-|J_i|} \leq 2^{d(1/a)} \) such \( u \). As a result the Taylor expansion is convergent at least \( 2^d - n2^{d(1/a)} = 2^d(1 - n/2^a) \) corners of \( [0,1]^d \).

For any \( \epsilon > 0 \) we have

\[
1 \leq s_z \leq 1 + \epsilon
\]

with overwhelming probability for \( z \sim U[0,1]^d \). For such \( z \)

\[
\nu(z) = f(x_i) + \sum_{i \not\in t} f(x_i) \prod_{j \in J_i} (1 - z_j) \quad (7)
\]

to within a factor of \( 1 + \epsilon \) and the right hand side of (7) is a function for which IGCS matches CS. Furthermore equation (7) holds exactly at the majority of the corners of \([0,1]^d\).

It is instructive to carry out the Taylor expansion to the next \( r = 1 \) term to see how a mismatch between CS and IGCS arises. Doing that we get

\[
\nu(z) \approx f(x_i) + \sum_{i \not\in t} f(x_i) \prod_{j \in J_i} (1 - z_j) - f(x_i) \sum_{i \not\in t} \prod_{j \in J_i} (1 - z_j)
\]

- \( \sum_{i \not\in t} f(x_i) \sum_{i' \not\in t} \prod_{j \in J_{i'}} (1 - z_j) \prod_{j \in J_i} (1 - z_j) \).

There are two new terms. The first new term also involves products over \( j \in J_i \) of \( 1 - z_j \) and so it also is one for which IGCS matches CS.

The second new term need not be of that form. It does however involve two such products of factors \( 1 - z_j \) and we know that each such product is typically smaller than \( \epsilon \). That term is then \( O(n^2 \epsilon^2) \) over most of \([0,1]^d\) while the other terms are \( O(n \epsilon) \). For large \( d \) it is easy to make \( n \epsilon \) negligible. For instance with \( \epsilon = 1/n^2 \) we get \( \Pr(z \in H_z) \leq n^\epsilon \exp(-|ad|/4) \). This is negligible for \( d \gg (16/a) \log(n) \).

The second new term is a weighted sum of

\[
g_{i,i'}(z) = \prod_{j \in J_i} (1 - z_j) \prod_{j \in J_{i'}} (1 - z_j) = \prod_{j \in J_i \cap J_{i'}} (1 - z_j) \times \prod_{j \in J_i \cup J_{i'}} (1 - z_j).
\]

It is the factors \( j \in J_i \cap J_{i'} \) that make IGCS differ from CS. The cohort Shapley value for \( g_{i,i'} \) is

\[
\frac{1}{|J_i \cup J_{i'}|} \times 1\{j \in J_i \cup J_{i'}\}.
\]
This follows from symmetry and the fact that \( g_{i,i'} \) only takes values 0 or 1 at the corners of the cube.

To find the IGCS value for \( g_{i,i'} \), note that

\[
g_{i,i'}(z) = \prod_{j \in J_i \cap J_i'} (1 - z_j)^2 \prod_{j \in J_i \triangle J_i'} (1 - z_j)
\]

where \( \triangle \) denotes the symmetric difference. For \( j' \in J_i \cap J_i' \)

\[
\frac{\partial}{\partial z_{j'}} g_{i,i'}(z) = -2 \prod_{j \in J_i \cap J_i' \setminus \{j'\}} (1 - z_j)^2 \prod_{j \in J_i \triangle J_i' \cup \{j'\}} (1 - z_j), \text{ so}
\]

\[
\frac{\partial}{\partial z_{j'}} g_{i,i'}(z1) = -2(1 - z)^{2|J_i \cap J_i'| + |J_i \triangle J_i'| - 1}
\]

which has IGCS value

\[
\frac{2}{2|J_i \cap J_i'| + |J_i \triangle J_i'|}.
\]

Similarly for \( j' \in J_i \triangle J_i' \)

\[
\frac{\partial}{\partial z_{j'}} g_{i,i'}(z) = -1 \prod_{j \in J_i \cap J_i' \setminus \{j'\}} (1 - z_j)^2 \prod_{j \in J_i \triangle J_i' \cup \{j'\}} (1 - z_j), \text{ so}
\]

\[
\frac{\partial}{\partial z_{j'}} g_{i,i'}(z1) = -(1 - z)^{2|J_i \cap J_i'| + |J_i \triangle J_i'| - 1}
\]

which has IGCS value

\[
\frac{1}{2|J_i \cap J_i'| + |J_i \triangle J_i'|}.
\]

The consequence is that IGCS gives extra weight to \( j \in J_i \cup J_i' \) and less to \( j \in J_i \triangle J_i' \). We expect that any given variable \( j \) will be in the double weight grouping for some pairs \( i, i' \) and in the single weight grouping for others. A variable \( j \) where \( x_{ij} \) is rarely dissimilar to \( x_{tj} \) is less likely to get those double weightings.

The above argument has not used the full generality of Corollary 1. Approximation within the wider class of functions considered there can only reduce the differences \( |\psi_j - \phi_j| \) between IGCS and CS compared to the multilinear functions in our Taylor expansion.

## 5 Experiments

In this section we show the feasibility of IGCS by applying it to real-world datasets. We use a 16-dimensional example from high energy physics where we can compare CS to IGCS. Then we use a 1024-dimensional example from computational chemistry where CS is infeasible.

### 5.1 Comparison of IGCS to CS for CERN data

Here we compare the performance of IGCS to the Cohort Shapley [Mase et al. 2019] with a low dimensional dataset. We evaluate the XAI methods by the insertion and deletion tests [Petsiuk et al. 2018] [Hama et al. 2022] described in Section 3.3. We will consider both interventional and conditional ABC measures.
5.1.1 Dataset and Setup

The CERN Electron Collision Data [McCauley 2014] is a dataset about dielectron collision events at CERN. It includes continuous variables representing the momenta and energy of the electrons, as well as discrete variables for the charges of the electrons (±1: positrons or electrons). Only the data whose invariant mass of two electrons (or positrons) is between 2 and 110 GeV are included. We treat it as a regression problem to predict their invariant mass from the other 16 features. Our model for this problem is a neural network described in Appendix A.1.

To apply CS we must choose similarity functions $S_j(x_i)$. Define the range of feature $j$ to be $r_j = \max_i x_{ij} - \min_i x_{ij}$. We take $S_j(x_i)$ to be one if and only if $|x_{ij} - x_{tj}| \leq \delta_j r_j$. We set each $\delta_j = 0.1$ so that two values of variable $j$ are similar if they are within 10% of the range of the variable.

5.1.2 Comparison methods

As noted in Section 2, the supervised on-manifold method of Frye et al. (2020) is a whole framework of methods. It is not clear how to develop one for this specific application, or how to train it well as the model it uses combines $2^{1024}$ different submodels depending on what variables are missing. For the purposes of comparison we have modified the Gaussian kernel weight (GKW) method from Aas et al. (2021) to get a model-free method. This modified method is described in detail in Appendix A.3. It was necessary to make a modification so that the method could be computed using only observed values of $f(x_i)$ instead of unobserved hybrid points.

The uniqueness Shapley criterion of Seiler et al. (2021) is a Shapley value based on taking $\nu(u)$ to be $-\log_2$ of the cardinality of the cohort $C_u$. Then the Shapley values $\phi_j$ quantify the relative importance of the variables in separating observation $t$ from the others. When aggregated over all observations it becomes a weighted sum of conditional entropies. We include it as a comparison because it is computationally feasible and it allows us to quantify whether variables for which a similarity condition more strongly makes the observation unique might also be important for predicting model output. That is, a variable in which $x_t$ tends to be an outlier might, to some extent, also be a variable that strongly affects its prediction or its response or its residual.

The prediction function for the CERN data is a differentiable function of 16 continuously varying variables. That makes it possible to use the original integrated gradients method on the model output (but not on the residuals).

The IG method compares a target data point to a baseline point. This is different from CS which ranks variables by how they split the point’s response value from the set of all observations’ responses. For IG we took all of the 2000 points in turn as the target point $t$ and computed the IG values from it to the other 1999 points. All in all we computed $\binom{2000}{2}$ insertion and deletion ABCs. We do not compute ABCs for kernel SHAP (KS) because the cost of doing $\binom{2000}{2}$ KS computations is prohibitive. We have extensive experience running KS on this same data set [Hama et al. 2022]. Based on that experience we expect KS to perform better than IG, but not much better than IG which was always a close second to KS in those comparisons.

We also include random variable ordering. It is known [Hama et al. 2022] that under random ordering the expected value of ABC for either insertion or deletion is zero when $f$ is additive. Whether $f$ is additive or not, the expected sum of insertion and deletion ABCs is zero. The random ordering is different for each target point.
### Table 2: Mean insertion and deletion ABCs for 2000 of the CERN data points calculated on feature attribution to predicted values, rounded to three places. The standard errors are with respect to 2000 held-out points for all methods except IG where they are for $\binom{2000}{2}$ pairs of points.

| Test Mode | Method             | Mean   | Std. Error |
|-----------|--------------------|--------|------------|
| Insertion | Cohort Shapley     | 10.213 | 0.142      |
|           | IGCS               | 9.726  | 0.137      |
|           | Random             | −0.461 | 0.130      |
|           | empirical GKW      | 6.569  | 0.139      |
|           | Uniqueness Shapley | 1.243  | 0.178      |
|           | Integrated Gradients| 2.260 | 0.004      |
| Deletion  | Cohort Shapley     | 9.176  | 0.114      |
|           | IGCS               | 8.835  | 0.122      |
|           | Random             | 0.406  | 0.129      |
|           | empirical GKW      | 5.866  | 0.106      |
|           | Uniqueness Shapley | 3.117  | 0.164      |
|           | Integrated Gradients| 2.192 | 0.003      |

#### 5.1.3 Results for CERN data

Table 2 shows insertion and deletion ABCs for the prediction function $f$ on the CERN data. These are conditional ABCs as described in Section 3.3. We include some standard errors, with respect to the 2000 held out points, as a descriptive statistic. For IG it is a standard error of $\binom{2000}{2}$ ABCs. We see that CS works best but IGCS is close behind even though there are only 16 variables. We also see that the sum of insertion and deletion ABCs is very close to zero for random ordering, consistent with theory. The empirical GKW method gets just over half the ABC values that CS and IGCS get. It outperforms plain IG. That could be because IG is an interventional method and Table 2 reflects conditional (cohort) ABCs. Uniqueness Shapley gets a low score. However, that measure does not even use the $f(x_i)$ values and so it is interesting that it can partially identify variables which move the cohort mean using only a measure of how outlying the target point is in those variables.

Table 3 shows insertion and deletion ABCs for the residuals in the model. The ABCs are smaller than for the prediction itself because the model has fit well. Again we see highest ABCs for CS with IGCS a close second and empirical GKW at about half the ABC. It is not possible to use the original IG method here because the residuals are not a differentiable function of the predictors.

Our conditional ABCs can be considered unfair to interventional methods like IG because those methods favor variables that bring big changes in interventions but were scored by how much they moved the cohort mean. To address this, we computed some interventional ABCs and we report them in Table 4. For an interventional method we need to choose a set of baseline-target pairs $(x_b, x_t)$. This is different from our conditional method comparing $x_t$ to some average of all data. The pairs we select are those from the counterfactual policy in Hama et al. (2022). That policy chooses the pairs as follows. For a given target $x_t$ we select baselines $x_b$ with:

- $x_{bj} \neq x_{tj}$ for all $j \in [d]$ including both particles’ charges,
- $x_b$ has one of the 20 smallest such $\|x_i - x_t\|$ values, and
- it maximizes $|f(x_b) - f(x_t)|$ subject to the above.
Table 3: Mean insertion and deletion ABCs for 2000 of the CERN data points calculated on feature attribution to residuals, the difference between annotated values and predicted values.

| Test Mode | Method       | Mean  | Std. Error |
|-----------|--------------|-------|------------|
| Insertion | Cohort Shapley | 1.239 | 0.022      |
|           | IGCS         | 1.155 | 0.021      |
|           | Random       | -0.023| 0.017      |
|           | empirical GKW | 0.601 | 0.019      |
|           | Uniqueness Shapley | 0.108 | 0.023      |
| Deletion  | Cohort Shapley | 1.034 | 0.020      |
|           | IGCS         | 0.993 | 0.020      |
|           | Random       | 0.019 | 0.018      |
|           | empirical GKW | 0.527 | 0.017      |
|           | Uniqueness Shapley | 0.254 | 0.023      |

The rationale for that policy is given in Hama et al. (2022). Pairs of responses can differ by more than responses differ from the average and we have also selected pairs that differ greatly (giving something to explain) so it is not surprising that larger ABCs can be found for this counterfactual policy. Because we adopt this policy also for picking reference data in calculation of IG and KS, we are able to compute KS in this setting. It performed best in Hama et al. (2022). IG is nearly as good. Conditional methods like CS and IGCS do not score well on these interventional ABCs. The empirical GKW method is the best conditional method under this interventional scorecard.

5.2 Feasibility in Data with many features

In this section we show the feasibility of IGCS in a dataset with 1024 binary features from an experiment in chemoinformatics. Because the features are binary, it is not difficult to define similarity. We take \( S_j(x_i) = 1 \) if and only if \( x_{ij} = x_{tj} \). We cannot compute CS exactly for this problem, but we can relate the IGCS findings to some domain knowledge. We can compute CS by Monte Carlo. We find that the resulting estimates do not attain ABC values competitive with IGCS. We cannot compute IG because the variables are all binary.

5.2.1 Dataset and Setup

For one elementary task in chemoinformatics, we consider the estimation of “logP” that measures the preference of a molecule to dissolve in lipids versus water. This metric is one of the rough standards in medicine known as Lipinski’s rule of five where it is an aspect of ‘druglikeness’. We use DeepChem (Ramsundar et al., 2019) and RDKit (https://www.rdkit.org/), a framework for chemoinformatics, and the data are collected from ZINC15 (Sterling and Irwin, 2015). The response we use are annotated values that are not experimental values but are instead calculated as sums of contributions from fragments of molecules called Wildman-Crippen LogP values (Wildman and Crippen, 1999).

The input data for these models encodes the molecules via the Extended-Connectivity Fingerprints (ECFP) (Rogers and Hahn, 2010). Each fingerprint is a binary vector that represents local information around each atom in a given molecule. This local information is transformed by a
| Test Mode | Method          | Mean   | Std. Error |
|-----------|-----------------|--------|------------|
| Insertion | Cohort Shapley  | 5.807  | 0.169      |
|           | IGCS            | 6.077  | 0.171      |
|           | empirical GKW   | 8.759  | 0.179      |
|           | Kernel SHAP     | 18.535 | 0.215      |
|           | Integrated Gradients | 18.289 | 0.213      |
| Deletion  | Cohort Shapley  | 5.884  | 0.176      |
|           | IGCS            | 6.176  | 0.175      |
|           | empirical GKW   | 9.259  | 0.156      |
|           | Kernel SHAP     | 16.752 | 0.176      |
|           | Integrated Gradients | 16.315 | 0.173      |

Table 4: Mean insertion and deletion ABCs evaluated in interventional measures for 2000 of the CERN data points calculated on feature attribution to predicted values, rounded to three places. The scores in other XAI methods are reported in [Hama et al.](2022).

hash function and stored as a 1024-dimensional binary vector. Each of these 1024 bits represents whether one or more copies of the corresponding information is present (bit=1) or whether none exist at all (bit=0), i.e., the feature is absent. The 1024 features we study were derived from a hash of 45,033 original features. This means that there are hash collisions of about 44 original features for each bit in our data.

Before presenting our model, we describe two papers doing related work. They are both about using XAI on similar problems to the one we discuss.

When two structurally similar compounds have very different chemoinformatic properties, it is called an activity cliff. [Tamura et al.] (2021) considered blackbox prediction of activity cliffs. Their feature set is somewhat different from the one we use. ECFP4 is a variant of ECFP that focuses on nonlocal information. They concatenate some binary vectors from ECFP4 to get the features they use. They had more than 2000 features in all. They applied kernel SHAP, presumably with Monte Carlo sampling given the high-dimensional feature vectors, to their support vector machine model.

In their application, KS was unsatisfactory. Because it attributed a lot of importance to absence of features, the domain experts could not use the results. They found better (more interpretable) results with the Tanimoto index computed with variables derived from domain knowledge.

[Heberle et al.] (2022) compared Crippen values and results of TreeSHAP. They used the Morgan Fingerprint, which is comparable to ECFP, as input vectors. They confirmed that the TreeSHAP is “almost identical” to Crippen contributions “relative to their own maximum absolute value”, that is, after normalizing by the maximum attribution. For this reason, we also compare the IGCS values to assigned Crippen values as a very rough guideline in following.

The model we used to estimate the annotated logP values is described in Appendix A.2. This model is very shallow, containing just one hidden layer with 3035 neurons to avoid overfitting. It was fit using mean squared error loss. The data set we consider is made up of 2000 molecules from the test data, that is, molecules that were held out during fitting of the one layer model.
5.2.2 Individual Examples in Chemoinformatics

We chose one molecule at random from the test data, shown in Figure 1(a). After we discuss that molecule we will consider another molecule, the one for which the model had its largest error, shown in Figure 1(b). We call these ZINC10 and ZINC61 below, which are shortened versions of their IDs. The pentagonal structures shown near the left of each molecule are known as ‘aromatic rings’. We will refer to that structure below. The best-known molecule with an aromatic ring is benzene, which has a hexagonal aromatic ring composed of six Carbon atoms.

The randomly chosen molecule ZINC10 has annotated logP = 2.8777. The average logP over all 2000 molecules is 1.2194 so we are left to explain a logP difference of 1.6583. With IGCS we get $\sum_{j=1}^{1024} \hat{\psi}_j = 1.6201$. The discrepancy arises because of numerical quadrature used to compute the integrated gradient vector $\hat{\psi}$. The predicted logP for this molecule was 2.555 and we had an average prediction of 1.2059 over the 2000 molecules. This leaves $2.555 - 1.2059 = 1.3491$ to explain and we got $\sum_{j=1}^{1024} \hat{\psi}_j = 1.3033$.

The distribution of IGCS values is represented in the left panel of Figure 2. The most negative $\hat{\psi}_j$ for this molecule was $-0.0279$, for the annotated value and $-0.0310$ for the predicted value. Both of those were for feature 726 which was absent (bit = 0) for this molecule. Feature 726 was present in 955 of the 2000 molecules and absent in 1045 of them. Figure 3 shows IGCS values $\hat{\psi}_{726}(t)$ letting the target $t$ be any of the 2000 molecules. We see that very generally $\hat{\psi}_{726}$ for the true annotated values is quite close to that for the predicted values for this feature. Also, presence of the feature is strongly linked to increasing logP while absence is strongly linked to decreasing logP with perfect separation of the signs for both annotated and predicted values. The molecule ZINC10 is shown as the green circle. If bit 726 is present it typically represents the existence of a carbon chain of length three. Such a chain is denoted by ccc in the Simplified Molecular-Input Line-Entry System.
The contributions to the predicted value and the annotated value. Left: the molecule (a) in Figure 1. Right: the molecule (b) in Figure 1.

(SMILES) notation (Weininger, 1988), which we also use for some other chemical features below.

The second largest negative value for ZINC10 is $-0.0166$ for the annotated value, and $-0.0172$ for the predicted value. This comes from feature 935 that is present in ZINC10. That feature represents the presence of Nitrogen which is known to be a water-soluble part which thus negatively contributes to logP. The corresponding Crippen value is $-0.3187$, and in this molecule there are two fragments of them so the total contribution to the annotated value is twice this Crippen value. There are three N in Figure 1(a), and the one in the aromatic ring contributes $-0.3239$ to the Crippen value.

The largest positive $\hat{\psi}_j$ for ZINC10 is 0.0295 for the annotated value and 0.0293 for the predicted values. These come from feature 849, which is present in ZINC10, representing a C–H part neighboring to the Nitrogen and the Oxygen in the aromatic ring. The corresponding Crippen value is 0.1581.

As shown above, while the units of ECFP do not match to fragments where Crippen values are annotated and they cannot be interpreted as ground truth of XAI, the signs and rough magnitudes of Crippen values are good indicator of domain knowledge in this field and our results agree with them.

The second molecule, shown in Figure 1(b) has largest loss in the test dataset. We denote that molecule ‘ZINC61’ in this paper. Before we study the features that explain this discrepancy we remark on the features that contribute to ZINC61’s values. The two largest positive contributions to annotated value $\hat{\psi}_j$ are from features 726 and 64, the same as for ZINC10. They both come from the presence of carbon chains ccc in the aromatic ring. Because the neighboring information around those carbon chains is different, they are hashed into different features. The largest negative feature of IGCS for annotated values at ZINC61 is feature 875 with a zero bit (absence). Feature 875 also typically means the presence of another kind of carbon chain in the aromatic ring.

The second largest negative feature in ZINC61 feature 935. It has the largest negative IGCS
value among present features. This feature also appeared in ZINC10 as the second largest negative feature. It is about presence of N and the assigned Crippen value for this atom is $-0.3187$. The feature where predicted importance falls short of annotated importance by the greatest amount is feature 210. This feature is present in this molecule. Presence of feature 210 is quite rare in the data set because only 26 of the 2000 molecules have it present. Presence is also rare for many other features. While it looks like a very rare feature, this dataset is highly sparse; about $1/4$ of the features appear in 26 or fewer molecules. More precisely, feature 210 is the 251st ‘sparsest’ of the 1024 features. Feature 210 in this molecule counts the presence of CCNC(N)=O (in SMILES notation). The IGCS for annotated value is $0.03160$ and for predicted value is $0.00891$. The second most underestimated feature is number 955. It refers the presence of CNC[C@H](c)N in this molecule. The number of molecules whose bit 955 is one is 22, but only 4 of them corresponds to CNC[C@H](c)N, while the other 18 molecules have presence due to hash collisions. In this sense, the various contributions should be assigned to each of the features represented in bit 955 depending on their properties, but the limited hash length prevents it. From this viewpoint, one would have to increase the hash length of ECFP to resolve those differences.

One of the strengths of IGCS is that we can use it to identify which features are associated with a large loss between observed and predicted responses. See Figure 2(b). The vertical distance of each point from diagonal line in this figure gives a good indicator of loss SHAP, because the loss function is the usual MSE in this task. We see that the greatest source of underestimation for ZINC61 comes from the underestimation of contributions from present features. The origins of these differences can be from the sparsity of the counts of each feature over the molecules in test dataset. This kind of information must be exploited to obtain accurate models. However, there is no universal
### Table 5: The ABCs with different sampling number in CS and IGCS of ZINC15 data. The numbers in the parenthesis represent the standard error in 2000 data.

| Number of Samples | Insertion ABC (Std. error) | Deletion ABC (Std. error) | Time (sec./data) |
|-------------------|-----------------------------|---------------------------|-----------------|
| 2049              | 0.052 (0.016)               | 0.063 (0.016)             | 0.517           |
| 10000             | 0.028 (0.017)               | 0.037 (0.017)             | 1.387           |
| 100000            | 0.071 (0.016)               | 0.060 (0.017)             | 14.760          |
| 200000            | 0.106 (0.015)               | 0.114 (0.016)             | 29.154          |
| 300000            | 0.128 (0.014)               | 0.144 (0.015)             | 44.720          |
| 400000            | 0.146 (0.014)               | 0.169 (0.015)             | 59.618          |
| 500000            | 0.149 (0.014)               | 0.179 (0.015)             | 74.503          |
| 50 steps IGCS     | 0.318 (0.010)               | 0.637 (0.011)             | 0.516           |
| 200 steps IGCS    | 0.330 (0.010)               | 0.664 (0.011)             | 0.661           |

5.3 Number of Samples and ABC

In Section 5.1 we compared several methods in terms of ABC using the CERN data. The CS results in Section 5.1 could use exact CS because with only $d = 16$ variables the cost was manageable. In the DeepChem setting it is not possible to compute CS. We are left to choose between IGCS and Monte Carlo (MC) sampling. That can be done by sampling from the $d!$ permutations underlying one expression for Shapley value or from other methods that just sample from the $d^{2d-1}$ incremental values. Sampling permutations is most straightforward because it can use uniform (unweighted) sampling. Here we compare IGCS to Monte Carlo (MC) sampling via ABC. The IGCS values will have a bias because $\psi_j \neq \phi_j$. MC estimates $\hat{\phi}_j$ are unbiased but can have high variance.

A comparison is made in Table 5 and illustrated in Figure 4. The averages are over 2000 held out molecules. The error bars represent ±1 standard error. This is a standard error over the 2000 molecules not over the MC simulations. The results are compared to IGCS with two choices for the number of function values in the Riemann sum. The IGCS reference lines are surrounded by bands shading plus or minus one standard error (over 2000 molecules) of the IGCS values.

The MC estimates show a clear upward trend; as more MC samples are taken, the estimated ABCs trend steadily upwards. While the underlying MC $\hat{\phi}_j$ are unbiased, the ABC estimates apply a nonlinear variable ranking transformation to those estimates so that the MC based estimates ABC are biased. From Figure 4 it is evident that truly enormous MC sampling cost would be required for the MC based method to get ABC scores comparable to those of IGCS. MC sampling with about the same cost in time as IGCS using 50 gradient points to estimate the integral yields an insertion ABC of 0.052 compared to 0.149 with half a million Monte Carlo samples. When ABC methods are used to demonstrate a good ordering, we can see that MC can have severe difficulty.

The computation time is measured in a server with Intel(R) Xeon(R) Gold 6132 CPU @ 2.60GHz and Tesla V100-SXM2.
6 Conclusion

We have introduced a new variable importance measure, IGCS, based on Aumann-Shapley theory. Like CS it is model-free, but unlike CS it can easily scale to high dimensional settings without incurring exponential cost. It has a value function on \([0,1]^d\) that is essentially multilinear over all but a negligible portion of the unit cube when \(d \gg \log(n)\). We thus expect it to match CS in the \(d = 1024\) example we considered and we saw it have very nearly the same quality as CS in a \(d = 16\) example where we could compute both CS and IGCS.

This model free method is useful in settings where the prediction function \(f\) is not available either because it is proprietary or because it is prohibitively expensive. It can also be used on raw data or on residuals from a model.

Acknowledgement

This work was supported by the U.S. National Science Foundation grants IIS-1837931 and DMS-2152780 and by Hitachi, Ltd. We thank Benjamin Seiler and Masashi Egi for helpful comments.

References

Aas, K., Jullum, M., and Løland, A. (2021). Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. Artificial Intelligence, 298:103502.

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pages 2623–2631.
Angwin, J., Larson, J., Mattu, S., and Kirchner, L. (2016). Machine bias: there’s software used across the country to predict future criminals, and it’s biased against blacks.

Aumann, R. J. and Shapley, L. S. (1974). *Values of Non-Atomic Games*. Princeton University Press, Princeton, NJ.

Azadkia, M. and Chatterjee, S. (2021). A simple measure of conditional dependence. *The Annals of Statistics*, 49(6):3070–3102.

Chen, H., Lundberg, S. M., and Lee, S.-I. (2022). Explaining a series of models by propagating Shapley values. *Nature communications*, 13(1):1–15.

Frye, C., de Mijolla, D., Begley, T., Cowton, L., Stanley, M., and Feige, I. (2020). Shapley explainability on the data manifold. *arXiv preprint arXiv:2006.01272*.

Hama, N., Mase, M., and Owen, A. B. (2022). Deletion and insertion tests in regression models. *arXiv preprint arXiv:2205.12423*.

Heberle, H., Zhao, L., Schmidt, S., Wolf, T., and Heinrich, J. (2022). XSMILES: interactive visualization for molecules, SMILES and XAI scores. *Research Square preprint*.

Hooker, G. and Mentch, L. (2019). Please stop permuting features: An explanation and alternatives. *arXiv e-prints*, pages arXiv–1905.

Hooker, S., Erhan, D., Kindermans, P.-J., and Kim, B. (2018). A benchmark for interpretability methods in deep neural networks. *arXiv preprint arXiv:1806.10758*.

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

Laurent, B. and Massart, P. (2000). Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pages 1302–1338.

Lindeman, R. H., Merenda, P. F., and Gold, R. Z. (1980). *Introduction to bivariate and multivariate analysis*. Scott, Foresman and Company, Homewood, IL.

Lundberg, S. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*.

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

Mase, M., Owen, A. B., and Seiler, B. (2019). Explaining black box decisions by Shapley cohort refinement. Technical report, arXiv:1911.00467.

Mase, M., Owen, A. B., and Seiler, B. B. (2021). Cohort Shapley value for algorithmic fairness. Technical report, arXiv:2105.07168.

McCuaity, T. (2014). Events with two electrons from 2010. CERN Open Data Portal.
Owen, G. (1972). Multilinear extensions of games. *Management Science*, 18(5-part-2):64–79.

Petsiuk, V., Das, A., and Saenko, K. (2018). RISE: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*.

Ramsundar, B., Eastman, P., Walters, P., Pande, V., Leswing, K., and Wu, Z. (2019). *Deep Learning for the Life Sciences*. O’Reilly Media. [https://www.amazon.com/Deep-Learning-Life-Sciences-Microscopy/dp/1492039837](https://www.amazon.com/Deep-Learning-Life-Sciences-Microscopy/dp/1492039837).

Ramsundar, B., Liu, B., Wu, Z., Verras, A., Tudor, M., Sheridan, R. P., and Pande, V. (2017). Is multitask deep learning practical for pharma? *Journal of chemical information and modeling*, 57(8):2068–2076.

Rényi, A. (1959). On measures of dependence. *Acta Mathematica Hungarica*, 10(3-4):441–451.

Rogers, D. and Hahn, M. (2010). Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754.

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

Seiler, B. B., Mase, M., and Owen, A. B. (2021). What makes you unique? *arXiv preprint arXiv:2105.08013*.

Shapley, L. S. (1953). A value for n-person games. In Kuhn, H. W. and Tucker, A. W., editors, *Contribution to the Theory of Games II (Annals of Mathematics Studies 28)*, pages 307–317. Princeton University Press, Princeton, NJ.

Sterling, T. and Irwin, J. J. (2015). ZINC 15–ligand discovery for everyone. *Journal of chemical information and modeling*, 55(11):2324–2337.

Sun, Y. and Sundararajan, M. (2011). Axiomatic attribution for multilinear functions. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 177–178.

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

Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR.

Székely, G. J. and Rizzo, M. L. (2013). Energy statistics: A class of statistics based on distances. *Journal of statistical planning and inference*, 143(8):1249–1272.

Tamura, S., Jasil, S., Miyao, T., and Fujitsu, K. (2021). Interpretation of ligand-based activity cliff prediction models using the matched molecular pair kernel. *Molecules*, 26(16):4916.

Teso, S., Alkan, Ö., Stammer, W., and Daly, E. (2022). Leveraging explanations in interactive machine learning: An overview. *arXiv preprint arXiv:2207.14526*.

Weininger, D. (1988). SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36.
A Detailed Model Descriptions

This appendix provides some background details on the experiments conducted in this article.

A.1 CERN Electron Collision Data

The hyperparameters for the model used in Section 5.1.1 are given in Table 6. They were obtained from a hyperparameter search using Optuna from Akiba et al. (2019). Each intermediate layer is a parametric ReLU with dropout. The dropout ratio is common to all of the layers. The model is trained with Huber loss. The model is overall very accurate but the very highest values are systematically underestimated, as seen in Hama et al. (2022) which also used this model.

| Hyperparameter       | Value                  |
|----------------------|------------------------|
| Dropout Ratio        | 0.11604                |
| Learning Rate        | $1.9163 \times 10^{-4}$ |
| Number of Neurons    | [509–421–65–368–122–477] |
| Huber Parameter      | 1.0                    |

Table 6: Parameters of the MLP model for the CERN Electron Collision Data.

A.2 ZINC15 Data

The model used in Section 5.2 is trained with RobustMultitaskRegressor of Ramsundar et al. (2017) by three tasks annotated in ZINC15 in DeepChem. The model is trained with mean squared loss. The hyperparameters for the model are given in Table 7. The number of nodes in a hidden layer is determined from a search using Optuna from Akiba et al. (2019). The learning rate is exponentially decaying from 0.01 with a decay rate of 0.8 per 1000 steps.

| Hyperparameter                  | Value      |
|---------------------------------|------------|
| Dropout Ratio                   | 0.25       |
| Number of Neurons in a hidden layer | [3935]    |

Table 7: Parameters of the MLP model that estimate logP values in ZINC15 from ECFP vectors.
A.3 Empirical Gaussian Kernel Weight

In this subsection, we describe our modification of the Gaussian kernel weight (GKW) method that we used in Section 5. The Gaussian kernel weight method of Aas et al. (2021) is not model free. Here we describe a model-free adaptation of it.

The value function in GKW takes the form

\[
\nu(u) = \frac{\sum_{i=1}^{n} s_u(x_i, x_t) f(x_i, x_t, -u)}{\sum_{i=1}^{n} s_u(x_i, x_t)} \tag{8}
\]

for a weight function

\[
s_u(x_i, x_t) = \exp \left( -\frac{D_u^2(x_i, x_t)}{2\sigma^2} \right)
\]

where

\[
D_u(x_i, x_t) = \sqrt{\frac{(x_{i,u} - x_{t,u})^T \Sigma_{uu}^{-1} (x_{i,u} - x_{t,u})}{|u|}}.
\]

The function \(D_u\) is a scaled Mahalanobis distance from \(x_{iu}\) to \(x_{tu}\). It uses \(\Sigma_{uu}\) which is the \(|u| \times |u|\) covariance matrix among the \(x_{iu}\) values. The distances and weights are measured in data space, not in indicator space, so they are available for real-valued features but perhaps not for other features.

GKW has a parameter \(\sigma > 0\). The authors of Aas et al. (2021) recommend choosing it via AIC. That is expensive and they also report that a default such as \(\sigma = 0.1\) works well in practice.

Before modifying GKW we make two observations. First, the normalization by \(|u|\) within \(D_u\) is unusual and it means that the distance from \(x_{iu}\) to \(x_{tu}\) could actually decrease when we incorporate another variable, replacing \(u\) by \(u \cup \{j\}\). We are not able to say whether this is a strength or weakness of GKW, just that it is an interesting property. Second, the value \(\sigma = 0.1\) gave us some concern. It makes \(s_u(x_i, x_t)\) equal to \(\exp(-50S^2_u)\) where \(S^2_u\) is a standardized mean square difference. It seems that this would then put the vast majority of the weight on the single closest point to \(x_t\), namely \(x_t\) itself. To address this concern we tried a larger values of \(\sigma\) (1.0 and 10.0) and found them to perform worse than \(\sigma = 0.1\) in our adaptation and so we work with the recommended default.

The GKW formula (8) is of interventional type. It uses hybrid points that are not observed. This means it cannot be used in a model-free setting. It may also reference impossible input combinations. We replace the hybrid points in (8) by observed data points, getting

\[
\nu(u) = \frac{\sum_{i=1}^{n} s_u(x_i, x_t) f(x_i)}{\sum_{i=1}^{n} s_u(x_i, x_t)}. \tag{9}
\]

We are then able to derive Shapley values from this model-free measure.