Feature Removal Is A Unifying Principle For Model Explanation Methods

Ian C. Covert
University of Washington
Seattle, WA
icovert@uw.edu

Scott Lundberg
Microsoft Research
Redmond, WA
scott.lundberg@microsoft.com

Su-In Lee
University of Washington
Seattle, WA
suinlee@uw.edu

Abstract

Researchers have proposed a wide variety of model explanation approaches, but it remains unclear how most methods are related or when one method is preferable to another. We examine the literature and find that many methods are based on a shared principle of *explaining by removing*—essentially, measuring the impact of removing sets of features from a model. These methods vary in several respects, so we develop a framework for *removal-based explanations* that characterizes each method along three dimensions: 1) how the method removes features, 2) what model behavior the method explains, and 3) how the method summarizes each feature's influence. Our framework unifies 25 existing methods, including several of the most widely used approaches (SHAP, LIME, Meaningful Perturbations, permutation tests). Exposing the fundamental similarities between these methods empowers users to reason about which tools to use and suggests promising directions for ongoing model explainability research.

1 Introduction

The proliferation of black-box models has made machine learning (ML) explainability an increasingly important subject, and researchers have now proposed a wide variety of model explanation approaches [6, 9, 10, 26, 31, 32, 33, 44, 46, 51]. Despite progress in the field, the relationships and trade-offs among these methods have not been rigorously investigated, and researchers have not always formalized their fundamental ideas about how to interpret models [24]. This makes the literature difficult to navigate and raises questions about whether existing methods relate to human processes for explaining complex decisions [29, 30].

Here, we present a comprehensive framework that unifies a substantial portion of the model explanation literature. Our framework is based on the observation that many methods can be understood as *simulating feature removal* to quantify each feature's influence on a model. The intuition behind these methods is similar (depicted in Figure 1), but each one takes a slightly different approach to the removal operation: some replace features with neutral values [32, 51], others marginalize over a distribution of values [26, 42], and still others train separate models for each subset of features [23, 44]. These methods also vary in other respects, as we describe below.

We refer to this class of approaches as *removal-based explanations* and identify 25[1] existing methods that rely on the feature removal principle, including several of the most widely used methods (SHAP, LIME, Meaningful Perturbations, permutation tests). We then develop a framework that shows how each method arises from various combinations of three choices: 1) how the method removes features from the model, 2) what model behavior the method analyzes, and 3) how the method summarizes each feature's influence on the model. By characterizing each method in terms of three precise

[1] This total count does not include minor variations on the approaches we identified.

NeurIPS 2020 Workshop: ML Retrospectives, Surveys & Meta-Analyses (ML-RSA).

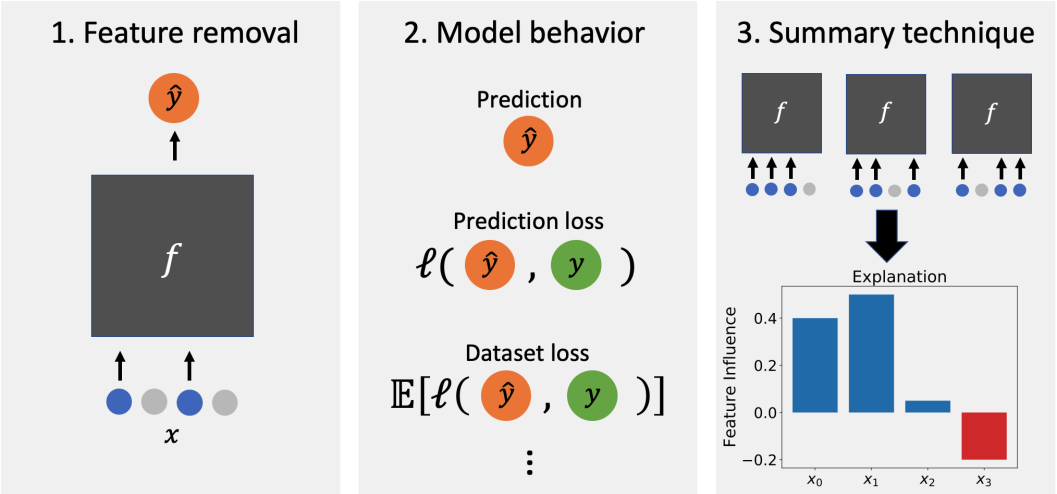

Figure 1: A unified framework for *removal-based explanations*. Each method is determined by three choices: how it removes features, what model behavior it analyzes, and how it summarizes feature influence.

mathematical choices, we are able to systematize their shared elements and reveal that they rely on the same fundamental approach—feature removal.

The model explanation field has grown significantly in the past decade, and we take a broader view of the literature than existing unification theories. Our framework's flexibility enables us to establish links between disparate classes of methods (e.g., computer vision-focused methods, global methods, game-theoretic methods, feature selection methods) and show that the literature is more interconnected than previously recognized. Exposing these underlying connections potentially raises questions about the degree of novelty in recent work, but we also believe that each method has the potential to offer unique advantages, either computationally or conceptually.

Through this work, we hope to empower users to reason more carefully about which tools to use, and we aim to provide researchers with new theoretical tools to build on in ongoing research. Our contributions include:

1. We present a framework that unifies 25 existing explanations methods. Our framework for **removal-based explanations** integrates classes of methods that were previously considered disjoint, including local and global approaches as well as feature attribution and feature selection methods.

2. We develop new mathematical tools to represent different approaches to removing features from brittle ML models. *Subset functions* and *model extensions* provide a common representation for various feature removal techniques, revealing that this choice is interchangeable between methods.

3. We generalize numerous explanation methods to express them within our framework, exposing connections that were not apparent in the original works. In particular, for several approaches we disentangle the substance of the methods from the approximations that make them practical.

We begin with background on the model explanation problem and a review of prior work (Section 2), and we then introduce our framework (Section 3). The next several sections examine our framework in detail by showing how it encompasses existing methods. Section 4 discusses how methods remove features, Section 5 formalizes the model behaviors analyzed by each method, and Section 6 describes each method's approach to summarizing feature influence. Finally, Section 7 concludes and discusses future research directions.

2 Background

Here, we introduce the model explanation problem and briefly review existing approaches and related unification theories.

2.1 Preliminaries

Consider a supervised ML model f that is used to predict a response variable $Y \in \mathcal{Y}$ using the input $X = (X_1, X_2, \ldots, X_d)$, where each X_i represents an individual feature, such as a patient's age. We use uppercase symbols (e.g., X) to denote random variables and lowercase ones (e.g., x) to denote their values. We also use $\mathcal{X}$ to denote the domain of the full feature vector X and $\mathcal{X}_i$ to denote the domain of each feature X_i. Finally, $x_S \equiv \{x_i : i \in S\}$ denotes a subset of features for $S \subseteq D \equiv \{1, 2, \ldots d\}$, and $\bar{S} \equiv D \setminus S$ represents a set's complement.

ML interpretability broadly aims to provide insight into how models make predictions. This is particularly important when f is a complex model, such as a neural network or a decision forest. The most active area of research in the field is *local interpretability*, which explains individual predictions, such as an individual patient diagnosis [26, 33, 46]; in contrast, *global interpretability* explains the model's behavior across the entire dataset [6, 10, 31]. Both problems are usually addressed using *feature attribution*, where a score is assigned to explain each feature's influence. However, recent work has also proposed the strategy of *local feature selection* [9], and other papers have introduced methods to isolate sets of relevant features [11, 14, 52].

Whether the aim is local or global interpretability, explaining the inner workings of complex models is fundamentally difficult, so it is no surprise that researchers keep devising new approaches. Commonly cited categories of approaches include perturbation-based methods [26, 51], gradient-based methods [39, 46], and inherently interpretable models [34, 53]. However, these categories refer to loose collections of approaches that seldom share a precise mechanism.

Besides the inherently interpretable models, virtually all of these approaches generate explanations by considering some class of perturbation to the input and using the outcomes to explain each feature's influence. Certain methods consider infinitesimal perturbations by calculating gradients [39, 40, 46, 48], but there are many possible perturbations [14, 26, 33, 51]. Our work is based on the observation that numerous perturbation strategies can be understood as simulating feature removal.

2.2 Related work

Prior work has made solid progress in exposing connections among disparate explanation methods. Lundberg and Lee proposed the unifying framework of *additive feature attribution methods* and showed that LIME, DeepLIFT, LRP and QII are all related to SHAP [5, 12, 26, 33, 37]. Similarly, Ancona et al. showed that Grad * Input, DeepLIFT, LRP and Integrated Gradients are all understandable as modified gradient backpropagations [4, 37, 46]. Most recently, Covert et al. showed that several global explanation methods can be viewed as *additive importance measures*, including permutation tests, Shapley Net Effects, and SAGE [6, 10, 23].

Relative to prior work, the unification we propose is considerably broader but nonetheless precise. As we describe below, our framework characterizes methods along three dimensions. The choice of how to remove features has been considered by many works [1, 3, 7, 15, 19, 20, 26, 28, 45]. The choice of what model behavior to analyze has been considered explicitly by only a few works [10, 25], as has the choice of how to summarize each feature's influence based on a set function [10, 12, 16, 26, 44]. To our knowledge, ours is the first work to consider all three dimensions simultaneously and unite them under a single framework.

Besides the methods that we focus on, there are also methods that do not rely on the feature removal principle. We direct readers to survey articles for a broader overview of the literature [2, 17].

3 Removal-Based Explanations

We now introduce our framework and briefly describe the methods it unifies.

Table 1: Choices made by existing removal-based explanations.

METHOD	REMOVAL	BEHAVIOR	SUMMARY
IME (2009)	Separate models	Prediction	Shapley value
IME (2010)	Marginalize (uniform)	Prediction	Shapley value
QII	Marginalize (marginals product)	Prediction	Shapley value
SHAP	Marginalize (conditional/marginal)	Prediction	Shapley value
KernelSHAP	Marginalize (marginal)	Prediction	Shapley value
TreeSHAP	Tree distribution	Prediction	Shapley value
LossSHAP	Marginalize (conditional)	Prediction loss	Shapley value
SAGE	Marginalize (conditional)	Dataset loss (label)	Shapley value
Shapley Net Effects	Separate models	Dataset loss (label)	Shapley value
Shapley Effects	Marginalize (conditional)	Dataset loss (output)	Shapley value
Permutation Test	Marginalize (marginal)	Dataset loss (label)	Remove individual
Conditional Perm. Test	Marginalize (conditional)	Dataset loss (label)	Remove individual
Feature Ablation (LOCO)	Separate models	Dataset loss (label)	Remove individual
Univariate Predictors	Separate models	Dataset loss (label)	Include individual
L2X	Missingness during training	Prediction mean loss	High-value subset
INVASE	Missingness during training	Prediction mean loss	High-value subset
LIME (Images)	Default values	Prediction	Linear model
LIME (Tabular)	Marginalize (replacement dist.)	Prediction	Linear model
PredDiff	Marginalize (conditional)	Prediction	Remove individual
Occlusion	Zeros	Prediction	Remove individual
CXPlain	Zeros	Prediction loss	Remove individual
RISE	Zeros	Prediction	Mean when included
MM	Default values	Prediction	Partitioned subsets
MIR	Extend pixel values	Prediction	High-value subset
MP	Blurring	Prediction	Low-value subset
EP	Blurring	Prediction	High-value subset
FIDO-CA	Generative model	Prediction	High-value subset

3.1 A unified framework

We develop a unified model explanation framework by connecting methods that define a feature's influence through the impact of removing it from a model. This perspective encompasses a substantial portion of the explainability literature: we find that 25 existing methods rely on this mechanism, including many of the most widely used approaches [6, 14, 26, 33].

These methods all remove groups of features from the model, but, beyond that, they take a diverse set of approaches. For example, LIME fits a linear model to an *interpretable representation* of the input [33], L2X selects the most informative features for a single example [9], and Shapley Effects examines how much of the model's variance is explained by each feature [31]. Perhaps surprisingly, their differences are easy to systematize because each method removes discrete sets of features.

As our main contribution, we introduce a framework that shows how these methods can be specified using only three choices.

Definition 1. Removal-based explanations *are model explanations that quantify the impact of removing sets of features from the model. These methods are determined by three choices:*

1. *(Feature removal) How the method removes features from the model (e.g., by setting them to default values or by marginalizing over a distribution of values)*
2. *(Model behavior) What model behavior the method analyzes (e.g., the probability of the true class or the model loss)*
3. *(Summary technique) How the method summarizes each feature's impact on the model (e.g., by removing a feature individually or by calculating the Shapley values)*

This precise yet flexible framework represents each choice as a specific type of mathematical function, as we show later. The framework unifies disparate explanation methods, and, by unraveling each method's choices, offers a step towards a better understanding of the literature by allowing explicit reasoning about the trade-offs among different approaches.

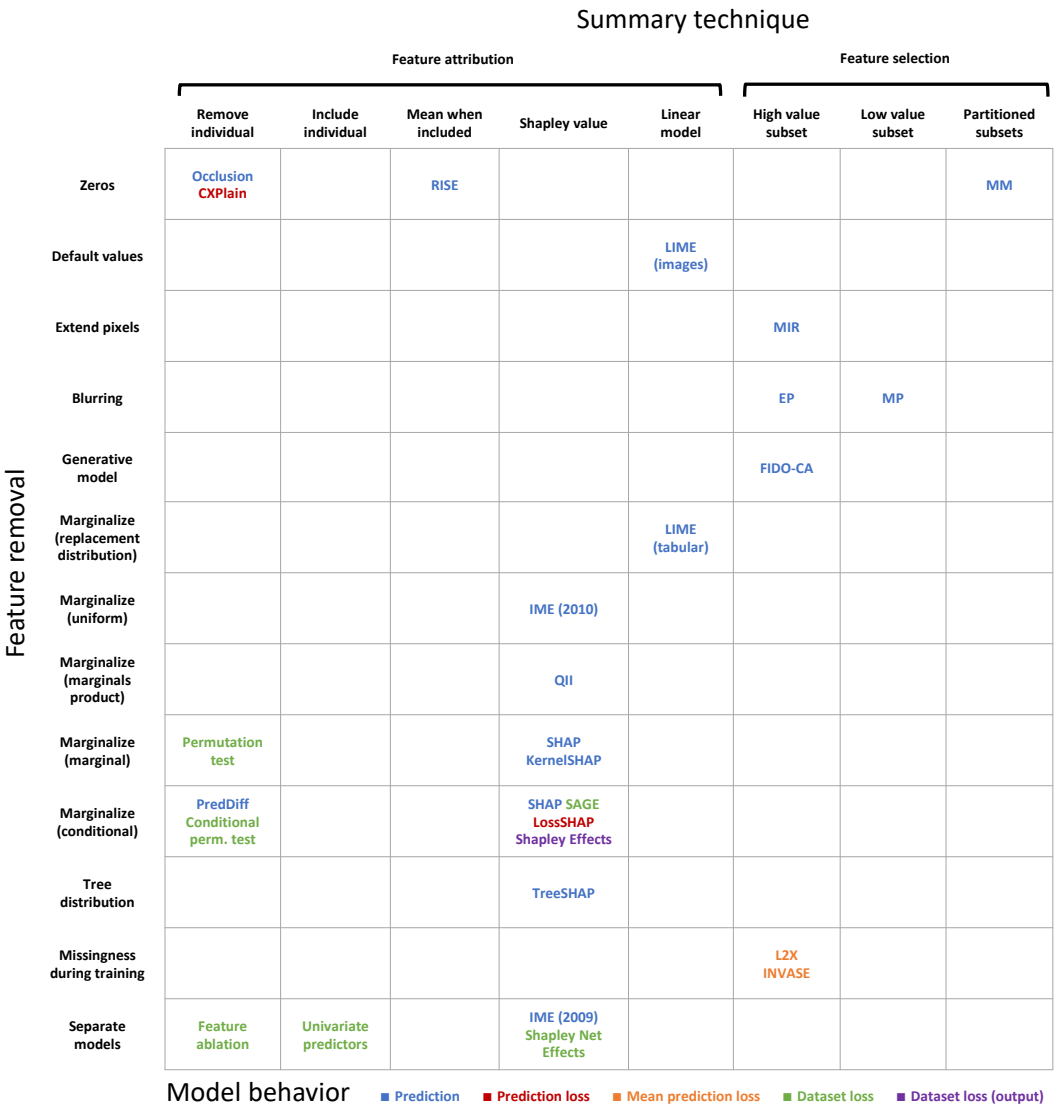

Figure 2: Visual depiction of the space of removal-based explanations.

3.2 Overview of existing approaches

We now outline some of our findings, which we present in more detail in the next several sections. In particular, we preview how existing methods fit into our framework and highlight groups of methods that appear similar in light of our feature removal perspective.

Table 1 lists the methods unified by our framework (with acronyms introduced in the next section). These methods represent diverse parts of the interpretability literature, including global methods [6, 31], computer vision-focused methods [14, 32, 51, 52], game-theoretic methods [10, 26, 43] and feature selection methods [9, 13, 49]. They are all unified by their reliance on feature removal.

Disentangling the details of each method shows that many approaches share one or more of the same choices. For example, most methods choose to explain individual predictions (the model behavior), and the most popular summary technique is the Shapley value [36]. These common choices raise important questions about how different these methods truly are and how their choices are justified.

To highlight similarities among the methods, we visually depict the space of removal-based explanations in Figure 2. Visualizing our framework reveals several regions in the space of methods that are crowded (e.g., methods that marginalize out removed features with their conditional distribution and that calculate Shapley values), while certain methods are relatively unique and spatially isolated (e.g.,

Table 2: Common combinations of choices in existing methods. Check marks (✓) indicate choices that are identical between methods.

REMOVAL	BEHAVIOR	SUMMARY	METHODS
	✓	✓	IME, QII, SHAP, KernelSHAP, TreeSHAP
✓		✓	SHAP, LossSHAP, SAGE, Shapley Effects
✓	✓		Occlusion, LIME (images), MM, RISE
	✓	✓	Feature ablation (LOCO), permutation tests, conditional permutation tests
✓	✓		Univariate predictors, feature ablation (LOCO), Shapley Net Effects
	✓	✓	SAGE, Shapley Net Effects
✓	✓		SAGE, conditional permutation tests
✓		✓	Shapley Net Effects, IME (2009)
✓		✓	Occlusion, CXPlain
	✓	✓	Occlusion, PredDiff
✓		✓	Conditional permutation tests, PredDiff
✓	✓		SHAP, PredDiff
✓	✓		MP, EP
	✓	✓	EP, FIDO-CA
✓	✓	✓	L2X, INVASE

RISE; LIME for tabular data; L2X and INVASE). Empty positions in the grid reveal opportunities to develop new methods; in fact, every empty position represents a viable new explanation method.

Finally, Table 2 shows groups of methods that differ in only one dimension of the framework. These methods are neighbors in the space of explanation methods (Figure 2), and it is remarkable how many instances of neighboring methods exist in the literature. Certain methods even have neighbors along every dimension of the framework (e.g., SHAP, SAGE, Occlusion, PredDiff, conditional permutation tests), reflecting how intimately connected the literature has become. The explainability literature is evolving and maturing, and our perspective provides a new approach for reasoning about the subtle relationships and trade-offs among existing approaches.

4 Feature Removal

Here, we define the mathematical tools necessary to remove features from ML models and then examine how existing explanation methods remove features.

4.1 Functions on subsets of features

Most ML models make predictions given a specific set of features $X = (X_1, \ldots, X_d)$. Mathematically, these models are functions are of the form $f : \mathcal{X} \mapsto \mathcal{Y}$, and we use $\mathcal{F}$ to denote the set of all such possible mappings. The principle behind removal-based explanations is to remove certain features to understand their impact on a model, but since most models require all the features to make predictions, removing a feature is more complicated than simply not giving the model access to it.

To remove features from a model, or to make predictions given a subset of features, we require a different mathematical object than $f \in \mathcal{F}$. Instead of functions with domain $\mathcal{X}$, we consider functions with domain $\mathcal{X} \times \mathcal{P}(D)$, where $\mathcal{P}(D)$ denotes the power set of $D \equiv \{1, \ldots, d\}$. To ensure invariance to the held out features, these functions must depend only on a set of features specified by a subset $S \in \mathcal{P}(D)$, so we formalize *subset functions* as follows.

Definition 2. A **subset function** *is a mapping of the form*

$$F : \mathcal{X} \times \mathcal{P}(D) \mapsto \mathcal{Y}$$

that is invariant to the dimensions that are not in the specified subset. That is, we have $F(x, S) = F(x', S)$ *for all* (x, x', S) *such that* $x_S = x'_S$*. We define* $F(x_S) \equiv F(x, S)$ *for convenience because the held out values* $x_{\bar{S}}$ *are not used by* F*.*

A subset function's invariance property is crucial to ensure that only the specified feature values determine the function's output, while guaranteeing that the other feature values do not matter. Another way of viewing subset functions is that they simulate the presence of missing data. While we use $\mathcal{F}$ to represent standard prediction functions, we use $\mathfrak{F}$ to denote the set of all possible subset functions.

We introduce subset functions here because they help conceptualize how different methods remove features from ML models. Removal-based explanations typically begin with an existing model $f \in \mathcal{F}$, and in order to quantify each feature's influence, they must establish a convention for removing it from the model. A natural approach is to define a subset function $F \in \mathfrak{F}$ based on the original model f. To formalize this idea, we define a model *extension* as follows.

Definition 3. *An* **extension** *of a model* $f \in \mathcal{F}$ *is a subset function* $F \in \mathfrak{F}$ *that agrees with* f *in the presence of all features. That is, the model* f *and its extension* F *must satisfy*

$$F(x) = f(x) \quad \forall x \in \mathcal{X}.$$

As we show next, extending an existing model is the first step towards specifying a removal-based explanation method.

4.2 Removing features from machine learning models

Existing methods have devised numerous ways to evaluate models while withholding groups of features. Although certain methods use different terminology to describe their approaches (e.g., deleting information, ignoring features, using neutral values, etc.), the goal of these methods is to measure a feature's influence through the impact of removing it from the model. Most proposed techniques can be understood as extensions $F \in \mathfrak{F}$ of an existing model $f \in \mathcal{F}$ (Definition 3).

We now examine each method's approach (see Appendix A for more details):

- (Zeros) Occlusion [51], RISE [32] and causal explanations (CXPlain) [35] remove features simply by setting them to zero:

$$F(x_S) = f(x_S, 0). \tag{1}$$

- (Default values) LIME for image data [33] and the Masking Model method (MM) [11] remove features by setting them to user-defined default values (e.g., gray pixels for images). Given default values $r \in \mathcal{X}$, these methods calculate

$$F(x_S) = f(x_S, r_{\bar{S}}). \tag{2}$$

  This is a generalization of the previous approach, and in some cases features may be given different default values (e.g., their mean).

- (Missingness during training) Learning to Explain (L2X) [9] and Instance-wise Variable Selection (INVASE) [49] use a model that has missingness introduced at training time. Removed features are replaced with zeros so that the model makes the following approximation:

$$F(x_S) = f(x_S, 0) \approx p(Y|X_S = x_S). \tag{3}$$

  This approach differs from Occlusion and RISE because the model is trained to recognize zeros as missing values rather than zero-valued features. A model trained with a loss function other than cross entropy loss would approximate a different quantity (e.g., the conditional expectation $\mathbb{E}[Y|X_S = x_S]$ for MSE loss).

- (Extend pixel values) Minimal image representation (MIR) [52] removes features in images by extending the values of neighboring pixels. This effect is achieved through a gradient-space manipulation.
- (Blurring) Meaningful Perturbations (MP) [14] and Extremal Perturbations (EP) [13] remove features from images by blurring them with a Gaussian kernel. This approach is *not* an extension of f because the blurred image retains dependence on the removed features. Blurring fails to remove large, low frequency objects (e.g., mountains), but it provides an approximate way to remove information from images.
- (Generative model) FIDO-CA [7] removes feature by replacing them with a sample from a conditional generative model (e.g. [50]). The held out features are drawn from a generative model represented by $p_G(X_{\bar{S}}|X_S)$, or $\tilde{x}_{\bar{S}} \sim p_G(X_{\bar{S}}|X_S)$ and predictions are made as follows:

$$F(x_S) = f(x_S, \tilde{x}_{\bar{S}}). \tag{4}$$

- (Marginalize with conditional) SHAP [26], LossSHAP [25] and SAGE [10] present a strategy for removing features by marginalizing them out using their conditional distribution $p(X_{\bar{S}} \mid X_S = x_S)$:

$$F(x_S) = \mathbb{E}\big[f(X) \mid X_S = x_S\big]. \tag{5}$$

  This approach is computationally challenging in practice, but recent work tries to achieve close approximations [1, 15]. Shapley Effects [31] implicitly uses this convention to analyze function sensitivity, while conditional permutation tests [42] and Prediction Difference Analysis (PredDiff) [54] do so to remove individual features.
- (Marginalize with marginal) KernelSHAP (a practical implementation of SHAP) [26] removes features by marginalizing them out using their joint marginal distribution $p(X_{\bar{S}})$:

$$F(x_S) = \mathbb{E}\big[f(x_S, X_{\bar{S}})\big]. \tag{6}$$

  This is the default behavior in SHAP's implementation,[2] and recent work discusses the benefits of this approach [20]. Permutation tests [6] use this approach to remove individual features from a model.
- (Marginalize with product of marginals) Quantitative Input Influence (QII) [12] removes held out features by marginalizing them out using the product of the marginal distributions $p(X_i)$:

$$F(x_S) = \mathbb{E}_{\prod_{i \in D} p(X_i)}\big[f(x_S, X_{\bar{S}})\big]. \tag{7}$$

- (Marginalize with uniform) The updated version of the Interactions Method for Explanation (IME) [43] removes features by marginalizing them out with a uniform distribution over the feature space. If we let $u_i(X_i)$ denote a uniform distribution over $\mathcal{X}_i$ (with extremal values defining the boundaries for continuous features), then features are removed as follows:

$$F(x_S) = \mathbb{E}_{\prod_{i \in D} u_i(X_i)}\big[f(x_S, X_{\bar{S}})\big]. \tag{8}$$

- (Marginalize with replacement distributions) LIME for tabular data replaces features with independent draws from *replacement distributions* (our term), each of which depends on the original feature values. When a feature X_i with value x_i is removed, discrete features are drawn from the distribution $p(X_i \mid X_i \neq x_i)$; when quantization is used for continuous features (LIME's default behavior[3]), continuous features are simulated by first generating a different quantile and then simulating from a truncated normal distribution within that bin. If we denote each feature's replacement distribution given the original value x_i as $q_{x_i}(X_i)$, then LIME for tabular data removes features as follows:

$$F(x, S) = \mathbb{E}_{\prod_{i \in D} q_{x_i}(X_i)}\big[f(x_S, X_{\bar{S}})\big]. \tag{9}$$

  Although this function F agrees with f given all features, it is *not* an extension because it does not satisfy the invariance property for subset functions.

[2] `https://github.com/slundberg/shap`

[3] `https://github.com/marcotcr/lime`

- (Tree distribution) Dependent TreeSHAP [25] removes features using the distribution induced by the model, which roughly approximates the conditional distribution. When splits for removed features are encountered in the model's trees, TreeSHAP averages predictions from the multiple paths in proportion to how often the dataset follows each path.
- (Separate models) Shapley Net Effects [23] and the original version of IME [44] are not based on a single model f but rather on separate models trained for each subset, which we denote as $\{f_S : S \subseteq D\}$. The prediction for a subset of features is given by that subset's model:

$$F(x_S) = f_S(x_S). \tag{10}$$

  Although this approach is technically an extension of the model f_D trained with all features, its predictions given subsets of features are not based on f_D. Similarly, feature ablation, also known as leave-one-covariate-out (LOCO) [22], trains models to remove individual features, and the univariate predictors approach (used mainly for feature selection) uses models trained with individual features [18].

Most of these approaches are extensions of an existing model f, so our formalisms provide useful tools for understanding how removal-based explanations remove features from models. However, consider two exceptions: the blurring technique (MP and EP) and LIME's approach with tabular data. Both provide functions of the form $F : \mathcal{X} \times \mathcal{P}(D) \mapsto \mathcal{Y}$ that agree with f given all features, but that still exhibit dependence on removed features. Based on our mathematical characterization of subset functions and their invariance to held out features, we argue that these two approaches do not fully remove features from the model. We conclude that the first dimension of our framework amounts to choosing an extension $F \in \mathfrak{F}$ of the model $f \in \mathcal{F}$.

5 Explaining Different Model Behaviors

Removal-based explanations all aim to demonstrate how a model works, but they can do so by analyzing a variety of model behaviors. We now consider the various choices of target quantities to observe as different features are withheld from the model.

The feature removal principle is flexible enough to explain virtually any function. For example, methods can explain a model's prediction, a model's loss function, a hidden layer in a neural network, or any node in a computation graph. In fact, removal-based explanations need not be restricted to the ML context: any function that accommodates missing inputs can be explained via feature removal by examining either its output or some function of its output as groups of inputs are removed. This perspective shows the broad potential applications for removal-based explanations.

However, since our focus is the ML context, we proceed by examining how existing methods work. Each method's target quantity can be understood as a function of the model output, which is represented by a subset function $F(x_S)$. Many methods explain the model output or a simple function of the output, such as the log-odds ratio. Other methods take into account a measure of the model's loss, for either an individual input or the entire dataset. Ultimately, as we show below, each method generates explanations based on a set function of the form

$$u : \mathcal{P}(D) \mapsto \mathbb{R},$$

which represents a value associated with each subset of features $S \subseteq D$. This set function represents the model behavior that a method is designed to explain.

We now examine the specific choices made by existing methods (see Appendix A for further details on each method). The various model behaviors that methods analyze, and their corresponding set functions, include:

- (Prediction) Occlusion, RISE, PredDiff, MP, EP, MM, FIDO-CA, MIR, LIME, SHAP (including KernelSHAP and TreeSHAP), IME and QII all analyze a model's prediction for an individual input $x \in \mathcal{X}$:

$$v_x(S) = F(x_S). \tag{11}$$

These methods quantify how holding out different features makes an individual prediction either higher or lower. For multi-class classification models, methods often use a single output that corresponds to the class of interest, and they can also apply a simple function to the model's output (for example, using the log-odds ratio rather than classification probability).

- (Prediction loss) LossSHAP and CXPlain take into account the true label y for an input x and calculate the prediction loss using a loss function ℓ:

$$v_{xy}(S) = -\ell\big(F(x_S), y\big). \tag{12}$$

By incorporating label information, these methods quantify whether certain features make the prediction more or less correct. The minus sign is necessary to give the set function a higher value when more informative features are included.

- (Prediction mean loss) L2X and INVASE consider the expected loss for a given input x according to the label's conditional distribution $p(Y|X = x)$:

$$w_x(S) = -\mathbb{E}_{p(Y|X=x)}\Big[\ell\big(F(x_S), Y\big)\Big]. \tag{13}$$

By averaging the loss across the label's distribution, these methods highlight features that correctly predict what *could* have occurred, on average.

- (Dataset loss w.r.t. label) Shapley Net Effects, SAGE, feature ablation (LOCO), permutation tests and univariate predictors consider the expected loss across the entire dataset:

$$v(S) = -\mathbb{E}_{XY}\Big[\ell\big(F(X_S), Y\big)\Big]. \tag{14}$$

These methods quantify how much the model's performance degrades when different features are removed. This set function can also be viewed as the predictive power derived from sets of features [10]. Recent work has proposed a SHAP value aggregation scheme that can be considered a special case of this approach [15].

- (Dataset loss w.r.t. output) Shapley Effects considers the expected loss with respect to the full model output:

$$w(S) = -\mathbb{E}_{X}\Big[\ell\big(F(X_S), F(X)\big)\Big]. \tag{15}$$

Though related to the previous approach [10], Shapley Effects focuses on each feature's influence on the model output rather than on the model performance.

Each set function serves a distinct purpose in exposing a model's dependence on different features. The first three approaches listed above analyze the model's behavior for individual predictions (local explanations); the last two take into account the model's behavior across the entire dataset (global explanations). Although their aims differ, these set functions are all in fact related. Each builds upon the previous ones by accounting for either the loss or data distribution, and their relationships can be summarized as follows:

$$v_{xy}(S) = \ell\big(v_x(S), y\big) \tag{16}$$

$$w_x(S) = \mathbb{E}_{p(Y|X=x)}\big[v_{xY}(S)\big] \tag{17}$$

$$v(S) = \mathbb{E}_{XY}\big[v_{XY}(S)\big] = \mathbb{E}_{X}\big[w_X(S)\big] \tag{18}$$

$$w(S) = \mathbb{E}_{X}\Big[\ell\big(v_X(S), v_X(D)\big)\Big] \tag{19}$$

These relationships show that explanations based on one set function are in some cases related to explanations based on another. For example, Covert et al. showed that SAGE explanations are the expectation of explanations provided by LossSHAP [10]—a relationship reflected in Eq. 18.

Understanding these connections is possible only because our framework disentangles each method's choices rather than viewing each method as a monolithic algorithm. We conclude by reiterating that removal-based explanations can explain virtually any function, and that choosing what to explain amounts to selecting a set function $u : \mathcal{P}(D) \mapsto \mathbb{R}$ to represent the model's dependence on different sets of features.

6 Summarizing Feature Influence

The third choice for removal-based explanations is how to summarize each feature's influence on the model. We examine the various summarization techniques and then discuss their computational complexity and approximation approaches.

6.1 Explaining set functions

The set functions we used to represent a model's dependence on different features (Section 5) are complicated mathematical objects that are difficult to communicate fully due to the exponential number of feature subsets and underlying feature interactions. Removal-based explanations confront this challenge by providing users with a concise summary of each feature's influence.

We distinguish between two main types of summarization approaches: feature attributions and feature selections. Many methods provide explanations in the form of *feature attributions*, which are numerical scores $a_i \in \mathbb{R}$ given to each feature $i = 1, \ldots, d$. If we use $\mathcal{U}$ to denote the set of all functions $u : \mathcal{P}(D) \mapsto \mathbb{R}$, then we can represent feature attributions as mappings of the form $E : \mathcal{U} \mapsto \mathbb{R}^d$, which we refer to as *explanation mappings*. Other methods take the alternative approach of summarizing set functions with a set $S^* \subseteq D$ of the most influential features. We represent these *feature selection* summaries as explanation mappings of the form $E : \mathcal{U} \mapsto \mathcal{P}(D)$. Both approaches provide users with simple summaries of a feature's contribution to the set function.

We now consider the specific choices made by each method (see Appendix A for further details). For simplicity, we let u denote the set function each method analyzes. Surveying the various removal-based explanation methods, the techniques for summarizing each feature's influence include:

- (Remove individual) Occlusion, PredDiff, CXPlain, permutation tests and feature ablation (LOCO) calculate the impact of removing a single feature from the set of all features, resulting in the following attribution values:

$$a_i = u(D) - u(D \setminus \{i\}). \tag{20}$$

  Occlusion, PredDiff and CXPlain can also be applied with groups of features in image contexts.

- (Include individual) The univariate predictors approach calculates the impact of including individual features, resulting in the following attribution values:

$$a_i = u(\{i\}) - u(\{\}). \tag{21}$$

  This is essentially the reverse of the previous approach: while that approach removes individual features from the complete set, this one adds individual features to the empty set.

- (Linear model) LIME fits a regularized weighted linear model to a dataset of perturbed examples. In the limit of an infinitely large dataset, this process approximates the following attribution values:

$$a_1, \ldots, a_d = \underset{b_0, \ldots, b_d}{\arg\min} \sum_{S \subseteq D} \pi(S) \Big(b_0 + \sum_{i \in S} b_i - u(S) \Big)^2 + \Omega(b_1, \ldots, b_d). \tag{22}$$

  In this problem, π represents a weighting kernel and Ω is a regularization function that is often set to the ℓ_1 penalty to encourage sparse attributions [47]. Since this summary is based on an additive model, the learned coefficients $(a_1, \ldots, a_d)$ represent values associated with including each feature.

- (Mean when included) RISE determines feature attributions by sampling many subsets $S \subseteq D$ and then calculating the mean value when a feature is included. Denoting the distribution of subsets as $p(S)$ and the conditional distribution as $p(S \mid i \in S)$, the attribution values are defined as

$$a_i = \mathbb{E}_{p(S \mid i \in S)} \big[u(S) \big]. \tag{23}$$

  In practice, RISE samples the subsets $S \subseteq D$ by removing each feature i independently with probability p, using $p = 0.5$ in the original experiments [32].

- (Shapley value) Shapley Net Effects, IME, Shapley Effects, QII, SHAP (including KernelSHAP, TreeSHAP and LossSHAP) and SAGE all calculate feature attribution values using the Shapley value, which we denote as $a_i = \phi_i(u)$. Shapley values are the only attributions that satisfy a number of desirable properties [36].
- (Low-value subset) MP selects a small set of features S^* that can be removed to give the set function a low value. It does so by solving the following optimization problem:

$$S^* = \arg\min_S \; u(D \setminus S) + \lambda |S|. \tag{24}$$

  In practice, MP uses additional regularizers and solves a relaxed version of this problem (see Section 6.2).
- (High-value subset) MIR solves an optimization problem to select a small set of features S^* that alone can give the set function a high value. For a user-defined minimum value t, the problem is given by:

$$S^* = \arg\min_S \; |S| \quad \text{s.}t. \;\; u(S) \geq t. \tag{25}$$

  L2X and EP solve a similar problem but switch the terms in the constraint and optimization objective. For a user-defined subset size k, the optimization problem is given by:

$$S^* = \arg\max_S \; u(S) \quad \text{s.t.} \;\; |S| = k. \tag{26}$$

  Finally, INVASE and FIDO-CA solve a regularized version of the problem with a parameter $\lambda > 0$ controlling the trade-off between the subset value and subset size:

$$S^* = \arg\max_S \; u(S) - \lambda |S|. \tag{27}$$

- (Partitioned subsets) MM solves an optimization problem to partition the features into S^* and $D \setminus S^*$ while maximizing the difference in the set function's values. This approach is based on the idea that removing features to find a low-value subset (as in MP) and retaining features to get a high-value subset (as in MIR, L2X, EP, INVASE and FIDO-CA) are both reasonable approaches for identifying influential features. The problem is given by:

$$S^* = \arg\max_S \; u(S) - \lambda u(D \setminus S) - \gamma |S|. \tag{28}$$

  In practice, MM incorporates regularizers and monotonic link functions to enable a more flexible trade-off between $u(S)$ and $u(D \setminus S)$ (see Appendix A).

As this discussion shows, every removal-based explanation generates summaries of each feature's influence on the underlying set function. In general, a model's dependencies are too complex to communicate fully, so explanations must provide users with a concise summary instead. As noted, most methods we discuss generate feature attributions, but several others generate explanations by selecting the most important features. These feature selection explanations are essentially coarse attributions that assign binary importance rather than a real number.

Interestingly, if the high-value subset optimization problems solved by MIR, L2X, EP, INVASE and FIDO-CA were applied to the set function that represents the dataset loss (Eq. 18), they would resemble conventional global feature selection problems [18]. The problem in Eq. 26 determines the set of k features with maximum predictive power, the problem in Eq. 25 determines the smallest possible set of features that achieve the performance represented by t, and the problem in Eq. 27 uses a parameter λ to control the trade-off. Though not generally viewed as a model explanation approach, global feature selection serves an identical purpose of identifying highly predictive features.

We conclude by reiterating that the third dimension of our framework amounts to a choice of explanation mapping, which takes the form $E : \mathcal{U} \mapsto \mathbb{R}^d$ for feature attribution or $E : \mathcal{U} \mapsto \mathcal{P}(D)$ for feature selection. Our discussion so far has shown that removal-based explanations can be specified using three precise mathematical choices, as depicted in Figure 3. These methods, which are often presented in ways that make their connections difficult to discern, are constructed in a remarkably similar fashion.

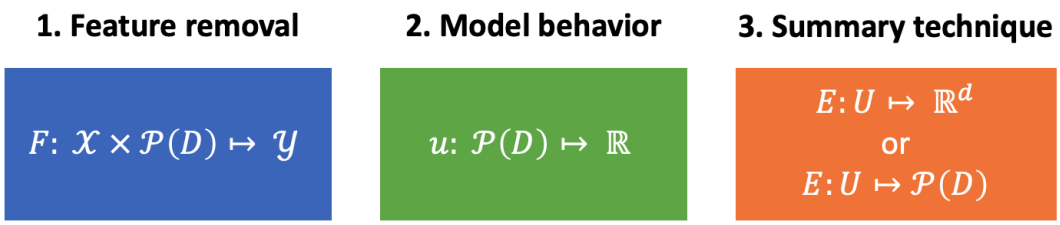

Figure 3: Removal-based explanations are specified by three precise mathematical choices: a subset function $F \in \mathfrak{F}$, a set function $u \in \mathcal{U}$, and an explanation mapping E (for feature attribution or selection).

6.2 Complexity and approximations

Showing how certain explanation methods fit into our framework requires distinguishing between their substance and the approximations that make them practical. Our presentation of these methods deviates from the original papers, which often focus on details of a method's implementation. We now bridge the gap by describing these methods' significant computational complexity and the approximations they use out of necessity.

The challenge with most summarization techniques described above is that they require calculating the underlying set function's value $u(S)$ for many subsets of features. In fact, without making any simplifying assumptions about the model or data distribution, several techniques must examine all 2^d subsets of features. This includes the Shapley value, RISE's summary technique and LIME's linear model. Finding exact solutions to several of the optimization problems (MP, MIR, MM, INVASE, FIDO-CA) also requires examining all subsets of features, and solving the constrained optimization problem (EP, L2X) for k features requires examining $\binom{d}{k}$ subsets, or $2^d d^{-\frac{1}{2}}$ subsets in the worst case.[4]

The only approaches with lower computational complexity are those that remove individual features (Occlusion, PredDiff, CXPlain, permutation tests, feature ablation) or include individual features (univariate predictors). These require only one subset per feature, or d total feature subsets.

Many summarization techniques have superpolynomial complexity in d, making them intractable for large numbers of features. However, these methods work in practice due to fast approximation approaches, and in some cases methods have even been devised to generate explanations in real-time. Strategies that yield fast approximations include:

- Attribution values that are the expectation of a random variable can be estimated using Monte Carlo approximations. IME [43], Shapley Effects [41] and SAGE [10] use sampling strategies to approximate Shapley values, and RISE also estimates its attributions via sampling [32].
- KernelSHAP and LIME are both based on linear regression models fitted to datasets containing an exponential number of datapoints. In practice, these techniques fit models to smaller sampled datasets, which means optimizing an approximate version of their objective function.
- TreeSHAP calculates Shapley values in polynomial time using a dynamic programming algorithm that exploits the structure of tree-based models. Similarly, L-Shapley and C-Shapley exploit the properties of models for structured data to provide fast Shapley value approximations [8].
- Several of the feature selection methods (MP, L2X, EP, MM, FIDO-CA) solve continuous relaxations of their discrete optimization problems. While these optimization problems could be solved by representing the set of features $S \subseteq D$ as a mask $m \in \{0, 1\}^d$, these methods instead use a mask variable of the form $m \in [0, 1]^d$. When these methods incorporate a penalty on the subset size $|S|$, they often use the convex relaxation $||m||_1$.

[4] This can be seen by applying Stirling's approximation to $\binom{d}{d/2}$ as d becomes large.

- One feature selection method (MIR) uses a greedy optimization algorithm. MIR determines a set of influential features $S \subseteq D$ by iteratively removing groups of features that do not reduce the predicted probability for the correct class.
- One feature attribution method (CXPlain) and three feature selection methods (L2X, INVASE, MM) generate real-time explanations by learning separate explainer models. CXPlain learns an explainer model using a dataset consisting of manually calculated explanations, which removes the need to iterate over each feature after training. L2X learns a model that outputs a set of features (represented by a k-hot vector) and INVASE learns a similar selector model that can output an arbitrary number of features; similarly, MM learns a model that outputs masks of the form $m \in [0, 1]^d$ for images. These techniques can be viewed as *amortized optimization* approaches [38] because they learn models that output approximate solutions in a single forward pass (similar to amortized inference [21]).

In conclusion, many methods provide efficient explanations despite using summarization techniques that are inherently intractable. Each approximation significantly speeds up computation relative to a brute-force calculation, but we predict that more approaches could be made to run in real-time by learning explainer models, as in the MM, L2X, INVASE and CXPlain approaches [9, 11, 35, 49].

7 Discussion

In this work, we developed a unified framework that characterizes a significant portion of the model explanation literature (25 existing methods). Removal-based explanations have a great degree of flexibility, and we systematized their differences by showing that each method is specified by three precise mathematical choices:

1. **How the method removes features.** Each method specifies a subset function $F \in \mathfrak{F}$ to make predictions with subsets of features, often based on an existing model $f \in \mathcal{F}$.
2. **What model behavior the method analyzes.** Each method implicitly relies on a set function $u : \mathcal{P}(D) \mapsto \mathbb{R}$ to represent the model's dependence on different groups of features. The set function describes the model's behavior either for an individual prediction or across the entire dataset.
3. **How the method summarizes each feature's influence.** Methods generate explanations that provide a concise summary of each feature's contribution to the set function $u \in \mathcal{U}$. Mappings of the form $E : \mathcal{U} \mapsto \mathbb{R}^d$ generate feature attribution explanations, and mappings of the form $E : \mathcal{U} \mapsto \mathcal{P}(D)$ generate feature selection explanations.

The growing interest in black-box ML models has spurred a remarkable amount of model explanation research, and in the past decade we have seen a number of publications proposing innovative new methods. However, as the field has matured we have also seen a growing number of unifying theories that reveal underlying similarities and implicit relationships [4, 10, 26]. Our framework for removal-based explanations is perhaps the broadest unifying theory yet, and it bridges the gap between disparate parts of the explainability literature.

An improved understanding of the field presents new opportunities for both explainability users and researchers. For users, we hope that our framework will allow for more explicit reasoning about the trade-offs between available explanation tools. The unique advantages of different methods are difficult to understand when they are viewed as monolithic algorithms, but disentangling their choices makes it simpler to reason about their strengths and weaknesses.

For researchers, our framework offers several promising directions for future work. We identify three key areas that can be explored to better understand the trade-offs between different removal-based explanations:

- Several of the methods characterized by our framework can be interpreted using ideas from information theory [9, 10]. We suspect that other methods can be understood with an information-theoretic perspective and that this may shed light on whether there are theoretically justified choices for each dimension of our framework.
- As we showed in Section 5, every removal-based explanation is based on an underlying set function that represents the model's behavior. Set functions can be viewed as *cooperative*

games, and we suspect that methods besides those that use Shapley values [10, 12, 26, 31, 44] can be related to techniques from cooperative game theory.

- Finally, it is remarkable that so many researchers have developed, with some degree of independence, explanation methods based on the same feature removal principle. We speculate that cognitive psychology may shed light on why this represents a natural approach to explaining complex decision processes. This would be impactful for the field because, as recent work has pointed out, explainability research is surprisingly disconnected from the social sciences [29, 30].

In conclusion, as the field evolves and the number of removal-based explanations continues to grow, we hope that our framework can serve as a foundation upon which future research can build.

A Method Details

Here, we provide additional details about some of the explanation methods discussed in the main text. In several cases, we presented generalized versions of methods that deviated from their explanations in the original papers.

A.1 Meaningful Perturbations (MP)

Meaningful Perturbations [14] considers multiple ways of deleting information from an input image, and the approach it recommends is a blurring operation. Given a mask $m \in [0, 1]^d$, MP uses a function $\Phi(x, m)$ to denote the modified input and suggests that the mask may be used to 1) set pixels to a constant value, 2) replace them with Gaussian noise, or 3) blur the image. In the blurring approach, each pixel x_i is blurred separately using a Gaussian kernel with standard deviation given by $\sigma \cdot m_i$ (for a user specified $\sigma > 0$).

To prevent adversarial solutions, MP incorporates a total variation norm on the mask, upsamples it from a low-resolution version, and uses a random jitter on the image during optimization. Additionally, MP uses a continuous mask $m \in [0, 1]^d$ in place of a binary mask $\{0, 1\}^d$ and the ℓ_1 penalty on the mask in place of the ℓ_0 penalty. Although MP's optimization tricks are key to providing visually compelling explanations, our presentation focuses on the most essential part of the optimization objective, which is reducing the classification probability while blurring only a small part of the image (Eq. 24).

A.2 Extremal Perturbations (EP)

Extremal Perturbations [13] is an extension of MP with several modifications. The first is switching the objective from a "removal game" to a "preservation game," which means learning a mask that retains rather than removes the salient information. The second is replacing the penalty on the subset size (or the mask norm) with a constraint. In practice, the constraint is enforced using a penalty, but the authors argue that it should still be viewed as a constraint due to the use of a large regularization parameter.

EP uses the same blurring operation as MP and introduces new tricks to ensure a smooth mask, but our presentation focuses on the most important part of the optimization problem, which is maximizing the classification probability while blurring a fixed portion of the image (Eq. 26).

A.3 FIDO-CA

FIDO-CA [7] is similar to EP but it replaces the blurring operation with features drawn from a generative model. The generative model p_G can condition on arbitrary subsets of features, and although its samples are non-deterministic, FIDO-CA achieves strong results using a single sample. The authors consider multiple generative models but recommend a generative adversarial network (GAN) that uses contextual attention [50]. The optimization objective is based on the same "preservation game" as EP, and the authors use the Concrete reparameterization trick [27] for optimization.

A.4 Minimal Image Representation (MIR)

The Minimal Image Representation approach [52] removes information from an image to determine which regions are salient for the desired class. MIR works by creating a segmentation of edges and regions and iteratively removing segments from the image (selecting those that least decrease the classification probability) until the remaining image is incorrectly classified. We view this as a greedy approach for solving the constrained optimization problem

$$\min_{S} \ |S| \quad \text{s.t.} \ \ u(S) \geq t,$$

where $u(S)$ represents the prediction with the specified subset of features and t represents the minimum allowable classification probability. Our presentation of MIR in the main text focuses on this view of the optimization objective rather than the specific greedy algorithm MIR uses (Eq. 25).

A.5 Masking Model (MM)

The Masking Model approach [11] observes that removing salient information (while preserving irrelevant information) and removing irrelevant information (while preserving salient information) are both reasonable approaches to understanding image classifiers. The authors refer to these tasks as discovering the smallest destroying region (SDR) and smallest sufficient region (SSR).

The authors adopt notation similar to MP [14], using $\Phi(x, m)$ to denote the transformation to the input given a mask $m \in [0, 1]^d$. For an input $x \in \mathcal{X}$, the authors aim to solve the following optimization problem:

$$\min_{m} \ \lambda_1 \mathrm{TV}(m) + \lambda_2 ||m||_1 - \log f\big(\Phi(x, m)\big) + \lambda_3 f\big(\Phi(x, 1 - m)\big)^{\lambda_4}.$$

The TV (total variation) and ℓ_1 penalty terms are both similar to MP and respectively encourage smoothness and sparsity in the mask. Unlike MP, MM learns a global explainer model that outputs approximate solutions to this problem in a single forward pass. In the main text, we provide a simplified presentation of the problem that does not include the logarithm in the third term or the exponent in the fourth term (Eq. 28). We view these as monotonic link functions that provide a more complex trade-off between the objectives but that are not necessary for finding informative solutions.

A.6 Learning to Explain (L2X)

The first theorem of the L2X paper [9] says that the explanation they seek is the distribution that optimizes the following objective:

$$\epsilon^*(x) = \underset{|S|=k}{\arg\min} \ -\mathbb{E}\big[\log p(Y \mid X_S = x_S) \mid X = x\big].$$

If we replace the conditional probability with a subset function $F(x_S) = p(Y|X_S = x_S)$ and allow for loss functions other than cross entropy, then we recover the version of this problem that we present in the main text. The L2X paper focuses on classification problems and an interpretation of their approach in terms of mutual information maximization; for a regression task evaluated with MSE loss, the approach could be interpreted analogously as performing conditional variance minimization.

A.7 Instance-wise Variable Selection (INVASE)

The INVASE method [49] is very similar to L2X, but it parameterizes the selector model differently. Rather than constraining the explanations to contain exactly k features, INVASE generates a set of features from a factorized Bernoulli distribution conditioned on the input $x \in \mathcal{X}$, using a regularization parameter $\lambda > 0$ to control the trade-off between the number of features and the expected value of the loss function. Instead of optimizing the selector model with reparameterization gradients, INVASE is learned using an actor-critic approach.

A.8 Prediction Difference Analysis (PredDiff)

Prediction Difference Analysis [54] removes individual features (or groups of features) and analyzes the difference in a model's prediction. Removed pixels are imputed by conditioning on their bordering pixels, which approximates sampling from the full conditional distribution $p(X_{\bar{S}}|X_S)$. Rather than measuring the prediction difference directly, the authors use attribution scores based on the log-odds ratio:

$$a_i = \log \frac{F(x)}{1 - F(x)} - \log \frac{F(x_{D \setminus \{i\}})}{1 - F(x_{D \setminus \{i\}})}.$$

We view this as another way of analyzing the difference in the model output for an individual prediction.

A.9 Causal Explanations (CXPlain)

CXPlain removes single features (or groups of features) for individual inputs and measures the change in the loss function [35]. The authors propose calculating the attribution values

$$a_i(x) = \ell\big(F(x_{D \setminus \{i\}}), y\big) - \ell\big(F(x, y)\big)$$

and then computing the normalized values

$$w_i(x) = \frac{a_i(x)}{\sum_{j=1}^{d} a_j(x)}.$$

The normalization step enables the use of a learning objective based on Kullback-Leibler divergence for the explainer model, which is ultimately used to calculate attribution values in a single forward pass. The authors explain that this approach is based on a "causal objective," but CXPlain is causal in the same sense as every other method described in our work.

A.10 Randomized Input Sampling for Explanation (RISE)

The RISE method [32] begins by generating a large number of randomly sampled binary masks. In practice, the masks are sampled by dropping features from a low-resolution mask independently with probability p, upsampling to get an image-sized mask, and then applying a random jitter. Due to the upsampling, the masks have values $m \in [0, 1]^d$ rather than $m \in \{0, 1\}^d$.

The mask generation process induces a distribution over the masks, which we denote as $p(m)$. The method then uses the randomly generated masks to obtain a Monte Carlo estimate of the following attribution values:

$$a_i = \frac{1}{\mathbb{E}[M_i]} \mathbb{E}_{p(M)}\big[f(x \odot M) \cdot M_i\big].$$

If we ignore the upsampling step that creates continuous mask values, we see that these attribution values are the mean prediction when a given pixel is included:

$$\begin{aligned} a_i &= \frac{1}{\mathbb{E}[M_i]} \mathbb{E}_{p(M)}\big[f(x \odot M) \cdot M_i\big] \\ &= \sum_{m \in \{0,1\}^d} f(x \odot m) \cdot m_i \cdot \frac{p(m)}{\mathbb{E}[M_i]} \\ &= \mathbb{E}_{p(M|M_i=1)}\big[f(x \odot M)\big]. \end{aligned}$$

A.11 Interactions Methods for Explanations (IME)

IME was presented in two separate papers [43, 44]. In the original version, the authors recommended training a separate model for each subset of features. In the second version, the authors proposed the more efficient approach of marginalizing out the removed features from a single model f.

The latter paper is ambiguous about the specific distribution used to marginalize out held out features [43]. Lundberg and Lee [26] view that features are marginalized out using their distribution from the training dataset (i.e., the marginal distribution). In contrast, Merrick and Taly [28] view IME as marginalizing out features using a uniform distribution. Upon a close reading of the paper, we opt for the uniform interpretation, but the specific interpretation of IME's choice of distribution does not impact any of our conclusions.

A.12 TreeSHAP

TreeSHAP uses a unique approach to handle held out features in tree-based models [25]. It accounts for missing features using the distribution induced by the underlying trees, and, since it exhibits no dependence on the held out features, it is a valid extension of the original model. However, it cannot be viewed as marginalizing out features using a simple distribution.

Given a subset of features, TreeSHAP makes a prediction separately for each tree and then combines each tree's prediction in the standard fashion. But when a split for an unknown feature is encountered, TreeSHAP averages predictions over the multiple paths in proportion to how often the dataset follows each path. This is similar but not identical to the conditional distribution because each time this averaging step is performed, TreeSHAP conditions only on coarse information about the features that preceded the split.

A.13 Shapley Net Effects

Shapley Net Effects was originally proposed for linear models that use MSE loss, but we generalize the method to arbitrary model classes and arbitrary loss functions. Unfortunately, Shapley Net Effects quickly becomes impractical with large numbers of features or non-linear models.

A.14 Shapley Effects

Shapley Effects analyzes a variance-based measure of a function's sensitivity to its inputs, with the goal of discovering which features are responsible for the greatest variance reduction in the model output [31]. The cooperative game described in the paper is:

$$u(S) = \mathrm{Var}\Big(\mathbb{E}\big[f(X) \mid X_S\big]\Big).$$

We present a generalized version to cast this method in our framework. In the appendix of Covert et al. [10], it was shown that this game is equal to:

$$\begin{aligned} u(S) &= \mathrm{Var}\Big(\mathbb{E}\big[f(X) \mid X_S\big]\Big) \\ &= \mathrm{Var}\big(f(X)\big) - \mathbb{E}\Big[\mathrm{Var}\big(f(X) \mid X_S\big)\Big] \\ &= c - \mathbb{E}\Big[\ell\big(\mathbb{E}\big[f(X) \mid X_S\big], f(X)\big)\Big] \\ &= c - \underbrace{\mathbb{E}\Big[\ell\big(F(X_S), f(X)\big)\Big]}_{\text{Dataset loss w.r.t. output}}. \end{aligned}$$

This derivation assumes that the loss function ℓ is MSE and that the subset function F is $F(x_S) = \mathbb{E}[f(X) \mid X_S = x_S]$. Rather than the original formulation, we present a cooperative game that is equivalent up to a constant value and that provides flexibility in the choice of loss function:

$$w(S) = -\mathbb{E}\Big[\ell\big(F(X_S), f(X)\big)\Big].$$

References

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

[2] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160, 2018.

[3] Chirag Agarwal and Anh Nguyen. Explaining an image classifier's decisions using generative models. *arXiv preprint arXiv:1910.04256*, 2019.

[4] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:1711.06104*, 2017.

[5] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS One*, 10(7):e0130140, 2015.

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

[7] Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. Explaining image classifiers by counterfactual generation. *arXiv preprint arXiv:1807.08024*, 2018.

[8] 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.

[9] Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan. Learning to explain: An information-theoretic perspective on model interpretation. *arXiv preprint arXiv:1802.07814*, 2018.

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

[11] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. In *Advances in Neural Information Processing Systems*, pages 6967–6976, 2017.

[12] 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.

[13] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2950–2958, 2019.

[14] 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.

[15] 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.

[16] Christopher Frye, Ilya Feige, and Colin Rowat. Asymmetric shapley values: incorporating causal knowledge into model-agnostic explainability. *arXiv preprint arXiv:1910.06358*, 2019.

[17] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Computing Surveys (CSUR)*, 51(5):1–42, 2018.

[18] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(Mar):1157–1182, 2003.

[19] Giles Hooker and Lucas Mentch. Please stop permuting features: An explanation and alternatives. *arXiv preprint arXiv:1905.03151*, 2019.

[20] Dominik Janzing, Lenon Minorics, and Patrick Blöbaum. Feature relevance quantification in explainable AI: A causality problem. *arXiv preprint arXiv:1910.13413*, 2019.

[21] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[22] Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.

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

[24] Zachary C Lipton. The mythos of model interpretability. *Queue*, 16(3):31–57, 2018.

[25] 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):2522–5839, 2020.

[26] 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.

[27] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[28] Luke Merrick and Ankur Taly. The explanation game: Explaining machine learning models with cooperative game theory. *arXiv preprint arXiv:1909.08128*, 2019.

[29] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

[30] Tim Miller, Piers Howe, and Liz Sonenberg. Explainable AI: Beware of inmates running the asylum or: How I learnt to stop worrying and love the social and behavioural sciences. *arXiv preprint arXiv:1712.00547*, 2017.

[31] Art B Owen. Sobol' indices and Shapley value. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):245–251, 2014.

[32] Vitali Petsiuk, Abir Das, and Kate Saenko. RISE: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

[33] 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.

[34] 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.

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

[36] Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

[37] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.

[38] Rui Shu. Amortized optimization, 2017.

[39] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[40] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[41] Eunhye Song, Barry L Nelson, and Jeremy Staum. Shapley effects for global sensitivity analysis: Theory and computation. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):1060–1083, 2016.

[42] Carolin Strobl, Anne-Laure Boulesteix, Thomas Kneib, Thomas Augustin, and Achim Zeileis. Conditional variable importance for random forests. *BMC Bioinformatics*, 9(1):307, 2008.

[43] Erik Štrumbelj and Igor Kononenko. An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, 11:1–18, 2010.

[44] 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.

[45] Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. *arXiv preprint arXiv:1908.08474*, 2019.

[46] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org, 2017.

[47] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[48] Shawn Xu, Subhashini Venugopalan, and Mukund Sundararajan. Attribution in scale and space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9680–9689, 2020.

[49] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. Invase: Instance-wise variable selection using neural networks. In *International Conference on Learning Representations*, 2018.

[50] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5505–5514, 2018.

[51] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014.

[52] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014.

[53] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.

[54] Luisa M Zintgraf, Taco S Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*, 2017.