Which Explanation Should I Choose? A Function Approximation Perspective to Characterizing Post hoc Explanations

Anonymous Author(s)
Affiliation
Address
email

Abstract

Despite the plethora of post hoc model explanation methods, the basic properties and behavior of these methods and the conditions under which each one is effective are not well understood. In this work, we bridge these gaps and address a fundamental question: Which explanation method should one use in a given situation? To this end, we adopt a function approximation perspective and formalize the *local function approximation (LFA) framework*. We show that popular explanation methods are instances of this framework, performing function approximations of the underlying model in different neighborhoods using different loss functions. We introduce a *no free lunch theorem for explanation methods* which demonstrates that no single method can perform optimally across all neighbourhoods and calls for choosing among methods. To choose among methods, we set forth a *guiding principle* based on the function approximation perspective, considering a method to be effective if it recovers the underlying model when the model is a member of the explanation function class. Then, we analyze the conditions under which popular explanation methods are effective and provide recommendations for choosing among explanation methods and creating new ones. Lastly, we empirically validate our theoretical results using various real world datasets, model classes, and prediction tasks. By providing a principled mathematical framework which unifies diverse explanation methods, our work characterizes the behaviour of these methods and their relation to one another, guides the choice of explanation methods, and paves the way for the creation of new ones.

1 Introduction

Machine learning models are increasingly being deployed in critical domains such as healthcare, law, and finance [12, 17, 42]. Consequently, there is a growing emphasis on explaining the predictions of these models to decision makers (e.g. doctors, judges) so that they can understand the rationale behind model predictions and determine when to rely on them. To this end, several methods have been proposed in literature to explain model predictions in a *post hoc* fashion. While these methods differ in a variety of ways, they can be broadly classified as *gradient-based* or *perturbation-based* methods based on their approach to estimating each feature's influence on the prediction. Popular state-of-the-art perturbation-based methods include `LIME` [32], `SHAP` [28] and `Occlusion`, and gradient-based ones include `Vanilla Gradients`, `Gradient x Input`, `SmoothGrad`, and `Integrated Gradients`. While these methods are applied to explain the predictions of complex models, their basic properties and behavior and the conditions under which each method is effective are not well understood. This understanding is particularly critical as different explanation methods have been known to generate disagreeing, and sometimes even contradictory, explanations [23].

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Prior works have taken the first steps towards characterizing explanation methods and establishing connections between them. For example, Agarwal et al. [2] proved that `C-LIME` (a continuous variant of `LIME` [32]) and `SmoothGrad` [35] converge to the same explanation in expectation. In addition, Lundberg and Lee [28] proposed a framework based on Shapley values to unify binary perturbation-based explanations, and Covert et al. [9] found that many perturbation-based methods share the property of estimating feature importance based on the change in model behavior upon feature removal. Ancona et al. [4] also analyzed four gradient-based explanation methods and the conditions under which they produce similar explanations. However, these analyses of explanation methods are based on mechanistic properties (such as Shapley values and feature removal), are limited in scope (connecting only two methods, only perturbation-based methods, or only gradient-based methods), and do not inform when one method is preferable to another.

To address the aforementioned gaps, we study explanation methods from a function approximation perspective and formalize an overarching mathematical framework that characterizes a broad set of methods (spanning both perturbation-based and gradient-based methods), and provides a principled approach to choose among these methods. Our work makes the following contributions:

1. We formalize the *linear function approximation (LFA) framework* and show that popular state-of-the-art perturbation-based and gradient-based explanation methods (such as `LIME`, `KernelSHAP`, `Occlusion`, `Vanilla Gradients`, `Gradient x Input`, `SmoothGrad`, and `Integrated Gradients`) are instances of this framework, essentially performing function approximations of the underlying model in different neighbourhoods using different loss functions.

2. We introduce a *no free lunch theorem for explanation methods* which demonstrates that no single explanation method can perform local function approximation faithfully across all neighbourhoods, thereby underscoring the need for a strategy to choose among various explanation methods.

3. To choose among methods, we set forth a *guiding principle* based on the function approximation perspective, deeming a method to be effective if its explanation model recovers the underlying model when the former is in the same model class as the latter (i.e. if the explanation model perfectly approximates the underlying model when it is able to do so).

4. We validate the theoretical results above through extensive empirical evaluation with various real-world datasets, predictive model classes, and prediction tasks.

2 Related Work

Post hoc explanations. Post hoc explanation techniques of existing pre-trained models can be classified based on their access to the complex model (i.e. black box vs. access to internals), scope of approximation (e.g. global vs. local), search technique (e.g. perturbation-based vs. gradient-based), and basic units of explanation (e.g. feature importance vs. rule-based). Examples are provided in the previous section. There has also been recent work on constructing *counterfactual explanations* which capture the changes that need to be made to a given instance in order to flip its prediction [41, 40, 18, 31, 27, 5, 19, 20]. Such explanations can be leveraged to provide recourse to individuals negatively impacted by algorithmic decisions. An alternate approach is to construct *global explanations*, summarizing the complete behavior of a black-box model by approximating it using an interpretable model [24, 6, 21].

Analyzing post hoc explanations. Recent work has shed light on the limitations of post hoc explanation methods. For instance, prior works [15, 34, 11, 1, 3] empirically demonstrated that perturbation-based methods, such as `LIME` and `SHAP`, may not be robust, i.e. small changes to the input may lead to drastic changes in the explanations. Adebayo et al. [1], Srinivas and Fleuret [36] also demonstrated that gradient-based methods do not always generate explanations that are faithful to the underlying model. In addition, recent research has also theoretically analyzed the robustness [25, 7] and other properties [14] of some popular post hoc explanation methods. However, none of these works provide a framework to characterize post hoc explanation methods and the connections between them or guide the selection of methods, all of which is the focus of our work.

3 Explanations as Local Function Approximations

In this section, we formalize the local function approximation framework and describe its connection to existing explanation methods. We proceed by defining the notation used in the rest of the paper.

Notation: Let $f : \mathcal{X} \to \mathcal{Y}$ be the black-box function we seek to explain in a post hoc manner, with input domain $\mathcal{X}$ (e.g. $\mathcal{X} = \mathbb{R}^d$ or $\{0,1\}^d$) and output domain $\mathcal{Y}$ (e.g. $\mathcal{Y} = \mathbb{R}$ or $[0,1]$). Let $\mathcal{G} = \{g : \mathcal{X} \to \mathcal{Y}\}$ be the class of interpretable models used to generate a local explanation for f by selecting a suitable interpretable model $g \in \mathcal{G}$.

We characterize locality around a point $\mathbf{x}_0 \in \mathcal{X}$ using a noise random variable ξ which is sampled from distribution $\mathcal{Z}$. Let $\mathbf{x}_\xi = \mathbf{x}_0 \oplus \xi$ be a perturbation of $\mathbf{x}_0$ generated by combining $\mathbf{x}_0$ and ξ using a binary operator $\oplus$ (e.g. addition, multiplication). Lastly, let $\ell(f, g, \mathbf{x}_0, \xi) \in \mathbb{R}^+$ be the loss function (e.g. MSE, cross-entropy) measuring the distance between f and g on a noise variable ξ around $\mathbf{x}_0$.

We now define the local function approximation framework.

Definition 1. ***Local function approximation*** *(LFA) of a black-box model f on a neighborhood distribution $\mathcal{Z}$ around $\mathbf{x}_0$ by an interpretable model family $\mathcal{G}$ and a loss function ℓ is given by*

$$g^* = \arg\min_{g \in \mathcal{G}} \mathop{\mathbb{E}}_{\xi \sim \mathcal{Z}} \ell(f, g, \mathbf{x}_0, \xi) \tag{1}$$

where a valid loss ℓ is such that $\mathbb{E}_{\xi \sim \mathcal{Z}}\, \ell(f, g, \mathbf{x}_0, \xi) = 0 \iff f(\mathbf{x}_\xi) = g(\mathbf{x}_\xi) \quad \forall \xi \sim \mathcal{Z}$

LFA is a formalisation of the function approximation perspective first introduced by `LIME` [32] to motivate local explanations. Note that the conceptual framework itself is distinct from the algorithm introduced by `LIME`. We elaborate on this distinction below.

(1) The LFA framework requires that f and g share the same input domain $\mathcal{X}$ and output domain $\mathcal{Y}$, a fundamental prerequisite for function approximation. This implies, for example, that using an interpretable model g with binary inputs ($\mathcal{X} = \{0,1\}^d$) to approximate a black-box model f with continuous inputs ($\mathcal{X} = \mathbb{R}^d$), as proposed in `LIME`, is not true function approximation.

(2) By imposing a condition on the loss function, the LFA framework ensures model recovery under specific conditions: g^* recovers f ($g^* = f$) through LFA when f itself is of the interpretable model class G ($f \in \mathcal{G}$) and perturbations span the input domain of f (domain$(\mathbf{x}) = \mathcal{X}$). This is a key distinction between the LFA framework and `LIME` (which has no such requirement) and guides the characterization of explanation methods in §4.

(3) Efficiently minimizing Equation 1 requires following standard machine learning methodology of splitting the perturbation data into train / validation / test sets and tuning hyper-parameters on validation data to ensure generalization. To our knowledge, implementations of `LIME` do not adopt this procedure, making it possible to overfit to a small number of perturbations.

The LFA framework is generic enough to accommodate a variety of explanation methods. In fact, we will show that specific instances of this framework converge to existing methods, as summarized in Table 1. At a high level, existing methods use a linear model g to locally approximate the black-box model f in different input domains (binary or continuous) around different local neighbourhoods specified by noise random variable ξ (where ξ is binary or continuous, drawn from a specified distribution, and combined additively or multiplicatively with point $\mathbf{x}_0$) using different loss functions (MSE or gradient-matching). We discuss the details of these connections in the following sections.

3.1 LFA with Continuous Noise: Gradient-Based Explanation Methods

To connect gradient-based explanation methods to the LFA framework, we leverage the gradient-matching loss function ℓ_{gm}. We define ℓ_{gm} and show that it is a valid loss function for LFA.

$$\ell_{gm}(f, g, \mathbf{x}_0, \xi) = \|\nabla_\xi f(\mathbf{x}_0 \oplus \xi) - \nabla_\xi g(\mathbf{x}_0 \oplus \xi)\|_2^2 \tag{2}$$

This loss function has been previously used in the contexts of generative modeling (where it is dubbed score-matching) [16] and model distillation [36]. However, to our knowledge, its use in interpretability is novel.

Table 1: Correspondence of existing explanation methods to instances of the LFA framework. Existing methods perform LFA of a black-box model f using the interpretable model class $\mathcal{G}$ of linear models where $g(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ over a local neighbourhood $\mathcal{Z}$ around point $\mathbf{x}_0$ based on a loss function ℓ. (Note: Exponential and Shapley kernels are defined in Appendix A.2.)

Explanation Method	**Local Neighborhood $\mathcal{Z}$ around $\mathbf{x}_0$**	**Loss Function ℓ**
C-LIME	$\mathbf{x}_0 + \xi;\ \xi(\in \mathbb{R}^d) \sim \text{Normal}(0, \sigma^2)$	MSE
SmoothGrad	$\mathbf{x}_0 + \xi;\ \xi(\in \mathbb{R}^d) \sim \text{Normal}(0, \sigma^2)$	Gradient Matching
Vanilla Gradients	$\mathbf{x}_0 + \xi;\ \xi(\in \mathbb{R}^d) \sim \text{Normal}(0, \sigma^2), \sigma \to 0$	Gradient Matching
Integrated Gradients	$\xi\mathbf{x}_0;\ \xi(\in \mathbb{R}) \sim \text{Uniform}(0, 1)$	Gradient Matching
Gradients $\times$ Input	$\xi\mathbf{x}_0;\ \xi(\in \mathbb{R}) \sim \text{Uniform}(a, 1), a \to 1$	Gradient Matching
LIME	$\mathbf{x}_0 \odot \xi;\ \xi(\in \{0,1\}^d) \sim$ Exponential kernel	MSE
KernelSHAP	$\mathbf{x}_0 \odot \xi;\ \xi(\in \{0,1\}^d) \sim$ Shapley kernel	MSE
Occlusion	$\mathbf{x}_0 \odot \xi;\ \xi(\in \{0,1\}^d) \sim$ Random one-hot vectors	MSE

Proposition 1. *The gradient-matching loss function ℓ_{gm} is a valid loss function for LFA up to a constant, i.e. $\mathbb{E}_{\xi\sim\mathcal{Z}}\, \ell_{gm}(f, g, \mathbf{x}_0, \xi) = 0 \iff f(\mathbf{x}_\xi) = g(\mathbf{x}_\xi) + C \quad \forall \xi \sim \mathcal{Z}$, where $C \in \mathbb{R}$.*

Proof. If $f(\mathbf{x}_\xi) = g(\mathbf{x}_\xi)$, then $\nabla_\xi f(\mathbf{x}_\xi) = \nabla_\xi g(\mathbf{x}_\xi)$ and it follows from the definition of ℓ_{gm} that $\ell_{gm} = 0$. Integrating $\nabla_\xi f(\mathbf{x}_\xi) = \nabla_\xi g(\mathbf{x}_\xi)$ gives $f(\mathbf{x}_\xi) = g(\mathbf{x}_\xi) + C$. □

Proposition 1 implies that, when using the linear model class $\mathcal{G}$ parameterized by $g(\mathbf{x}) = \mathbf{w}^\top\mathbf{x} + b$ to approximate f, g^* recovers $\mathbf{w}$ but not b. This can be fixed by setting $b = f(0)$.

Theorem 1. *LFA with gradient-matching loss is equivalent to (1) `SmoothGrad` for additive continuous Gaussian noise, which converges to `Vanilla Gradients` in the limit of a small standard deviation for the Gaussian distribution; and (2) `Integrated Gradients` for multiplicative continuous Uniform noise, which converges to `Gradient x Input` in the limit of a small support for the Uniform distribution.*

Proof Sketch. For `SmoothGrad` and `Integrated Gradients`, the idea is that these methods are exactly the first-order stationary points of the gradient-matching loss function under their respective noise distributions. For `Vanilla Gradients` and `Gradient x Input`, the result is derived by taking the specified limits and using the Dirac delta function. Full proofs are in Appendix A.2.

Along with gradient-based explanation methods, `C-LIME` [2] is an instance of the LFA framework by definition, using the MSE loss function. The analysis in this section characterizes methods that use continuous noise. It does not extend to binary nor discrete noise methods as gradients nor continuous random variables apply in these domains. In the next section, we discuss binary noise methods.

3.2 LFA with Binary Noise: LIME, KernelSHAP and Occlusion maps

Theorem 2. *LFA with multiplicative binary noise and MSE loss is equivalent to (1) LIME for noise sampled from an unnormalized exponential kernel over binary vectors; (2) KernelSHAP for noise sampled from an unnormalized Shapley kernel; and (3) Occlusion for noise in the form of one-hot vectors.*

Proof Sketch. For `LIME` and `KernelSHAP`, the equivalence is mostly by definition; we need only to account for the weighting kernel (which is not part of LFA). We show that the LFA framework using these kernels yields the respective explanation method in expectation via importance sampling. For `Occlusion`, the equivalence involves enumerating all perturbations and computing the resulting stationary points of LFA. Full proofs are in Appendix A.2.

4 When Do Explanations Perform Model Recovery?

Having described the LFA framework and its connections to existing explanation methods, we now leverage this framework to analyze the performance of methods under different conditions. We

introduce a *no free lunch theorem for explanation methods*, inspired by classical no free lunch theorems in learning theory and optimization. Then, we assess the ability of existing methods to perform *model recovery* based on which we then provide recommendations for choosing among methods.

4.1 No Free Lunch Theorem for Explanation Methods

An important implication of the function approximation perspective is that no explanation can be optimal across all neighborhoods for each explanation is designed to perform LFA in a specific neighborhood. We distill this intuitive observation into the following theorem.

Theorem 3 (No Free Lunch for Explanation Methods)**.** *Consider the scenario where we explain a black-box model f around point $\mathbf{x}_0$ using an interpretable model g from class $\mathcal{G}$ and a valid loss function ℓ where the distance between f and $\mathcal{G}$ is given by $d(f, \mathcal{G}) = \min_{g \in \mathcal{G}} \max_{\mathbf{x} \in \mathcal{X}} \ell(f, g, 0, \mathbf{x})$. Then, for any explanation g^* on a neighborhood distribution $\xi_1 \sim \mathcal{Z}_1$ such that $\max_{\xi_1} \ell(f, g^*, \mathbf{x}_0, \xi_1) \leq \epsilon$, we can always find another neighborhood $\xi_2 \sim \mathcal{Z}_2$ such that $\max_{\xi_2} \ell(f, g^*, \mathbf{x}_0, \xi_2) \geq d(f, \mathcal{G})$.*

Proof Sketch. The proof entails constructing an "adversarial" input for an explanation g^* such that g^* has a large loss for this input and then creating a neighborhood that contains this adversarial input which will provably have a large loss. The magnitude of this loss is $d(f, \mathcal{G})$, the distance between f and the model class $\mathcal{G}$, inspired by the Haussdorf distance. Full proof is in Appendix A.3.

Thus, an explanation on a finite $\mathcal{Z}_1$ necessarily cannot approximate function behaviour at all other points, especially when $\mathcal{G}$ is less expressive than f, which is indicated by a large value of $d(f, \mathcal{G})$. Thus, in the general case, one cannot perform model recovery as $\mathcal{G}$ is less expressive than f.

An important implication of Theorem 3 is that seeking to find the "best" explanation without specifying a corresponding neighborhood is futile as no universal "best" explanation exists. Furthermore, once the neighborhood is specified, the best explanation is exactly given by the corresponding instance of the LFA framework.

In the next section, we consider the special case when $d(f, \mathcal{G}) = 0$ (i.e. when $f \in \mathcal{G}$), where Theorem 3 does not apply as the same explanation can be optimal for multiple neighborhoods and model recovery is thus possible.

4.2 Characterizing Explanation Methods via Model Recovery

We proceed by formally stating the model recovery condition for explanation methods and then use this condition as a guiding principle to choose among methods.

Definition 2 (Model Recovery: Guiding Principle)**.** *Given an instance of the LFA framework with a black-box model f such that $f \in \mathcal{G}$ and a specific noise type (e.g. Gaussian, Uniform), an explanation method performs model recovery if there exists some noise distribution $\xi \sim \mathcal{Z}$ such that LFA returns $g^* = f$.*

In other words, when the black-box model f itself is of the interpretable model class G, there must exist some setting of the noise distribution (within the noise type specified in the instance of the LFA framework) that is able to recover the black-box model. (Note that model recovery is a consequence of the conditions specified on the loss function.) Thus, in this special case, we require *local function approximation* to lead to *global model recovery* over all inputs. This criterion can be thought of as a "sanity check" for explanation methods to ensure that they remain faithful to the black-box model.

Next, we analyze the impact of the choice of perturbation neighborhood $\mathcal{Z}$, the binary operator $\oplus$, and the interpretable model class $\mathcal{G}$ on an explanation method's ability to satisfy the model recovery guiding principle in different input domains $\mathcal{X}$. Note that while we can choose $\mathcal{Z}$, $\oplus$, and $\mathcal{G}$, we cannot choose $\mathcal{X}$ (the input domain determined by f).

Which methods should I use for continuous $\mathcal{X}$? We now analyze the model recovery properties of existing explanation methods when the input domain is continuous. We consider methods based on additive continuous noise (`SmoothGrad`, `Vanilla Gradients`, and `C-LIME`), multiplicative continuous noise (`Integrated Gradients` and `Gradient x Input`), and multiplicative binary noise (`LIME`, `KernelSHAP`, and `Occlusion`). For these methods, we make the following remark regarding model recovery for the class of linear models.

Remark 1. *For $\mathcal{X} = \mathbb{R}^d$ and linear models f and g where $f(\mathbf{x}) = \mathbf{w}_f^\top \mathbf{x}$ and $g(\mathbf{x}) = \mathbf{w}_g^\top \mathbf{x}$, additive noise methods recover f (i.e. $\boldsymbol{w}_g = \boldsymbol{w}_f$) while binary and multiplicative continuous noise methods do not and instead recover $\boldsymbol{w}_g = \boldsymbol{w}_f \odot \boldsymbol{x}$.*

This remark can be verified by directly evaluating the explanations (weights) of linear models, where the gradient exactly corresponds to the weights.

We point out that the inability of multiplicative continuous noise methods to recover the black-box model is not due to the multiplicative nature of the noise, but rather due to the parameterization of the loss function. Specifically, these methods (implicitly) use the loss function $\ell(f, g, \mathbf{x}_0, \xi) = \|\nabla_\xi f(\mathbf{x}_\xi) - \nabla_\xi g(\xi)\|_2^2$. Slightly changing the loss function to $\ell(f, g, \mathbf{x}_0, \xi) = \|\nabla_\xi f(\mathbf{x}_\xi) - \nabla_\xi g(\mathbf{x}_\xi)\|_2^2$, i.e. replacing $g(\xi)$ with $g(\mathbf{x}_\xi)$, would enable g^* to recover f. This would change `Integrated Gradients` to $\int_{\alpha=0}^{1} \nabla_{\alpha\mathbf{x}} f(\alpha\mathbf{x})$ (omitting the input multiplication term) and `Gradient x Input` to `Vanilla Gradients`.

A similar argument can be made for binary noise methods which parameterize the loss function as $\ell(f, g, \mathbf{x}_0, \xi) = \|f(\mathbf{x}_\xi) - g(\xi)\|^2$. By changing the loss function to $\ell(f, g, \mathbf{x}_0, \xi) = \|f(\mathbf{x}_\xi) - g(\mathbf{x}_\xi)\|^2$, binary noise methods can recover f for the case described in Remark 1. However, binary noise methods for continuous domains are unreliable, as there are cases where, despite the modification to ℓ, model recovery is not guaranteed. The following is an example of such a case.

Remark 2. *For $\mathcal{X} = \mathbb{R}^d$, periodic functions f and g where $f(\mathbf{x}) = \sum_{i=1}^{d} \sin(\mathbf{w}_{f_i} \odot \mathbf{x}_i)$ and $g(\mathbf{x}) = \sum_{i=1}^{d} \sin(\mathbf{w}_{g_i} \odot \mathbf{x}_i)$, and an integer n, binary noise methods do not perform model recovery for $|w_{f_i}| \geq \frac{n\pi}{\mathbf{x}_{0_i}}$.*

This is because, for the conditions specified, $\sin(\mathbf{w}_{f_i}\mathbf{x}_{0_i}) = \sin(\pm n\pi) = \sin(0) = 0$, i.e. $\sin(\mathbf{w}_{f_i}\mathbf{x}_{0_i})$ outputs zero for all binary perturbations, thereby preventing model recovery. In this case, the discrete nature of the noise makes model recovery impossible. In general, discrete noise is inadequate for the recovery of models with large frequency components.

Which methods should I use for binary $\mathcal{X}$? In the binary domain, continuous noise methods are invalid, restricting us to binary noise methods. By the discussion above, methods with perturbation neighborhoods characterized by multiplicative binary perturbations, such as `LIME`, `KernelSHAP`, and `Occlusion`, only enable g^* to recover f in the binary domain. Note that the sinusoidal example in Remark 2 does not apply in this regime due to the continuous nature of its domain.

Which methods should I use for discrete $\mathcal{X}$? In the discrete domain, continuous noise methods are also invalid. In addition, binary noise methods, such as `LIME`, `KernelSHAP` and `Occlusion`, also cannot be used as model recovery is not guaranteed in the sinusoidal case (Remark 2), following a logic similar to that presented for continuous noise. We notice that none of the existing methods in Table 1 perform general discrete perturbations, suggesting that these methods are not suitable for the discrete domain. Thus, in the discrete domain, a user can apply the LFA framework to define a new explanation method, specifying an appropriate discrete noise type. In the next section, we discuss more broadly about how one can use the LFA framework to create novel explanation methods.

4.3 Designing Novel Explanations with LFA

The LFA framework not only unifies existing explanation methods but also guides the creation of new ones. To explain a given black-box model prediction using the LFA framework, a user must specify the (1) interpretable model class $\mathcal{G}$, (2) neighborhood distribution $\mathcal{Z}$, (3) loss function ℓ, and (4) binary operator $\oplus$ to combine the input and the noise. Specifying these completely specifies an instance of the LFA framework, thereby generating an explanation method tailored to a given context.

To illustrate this, consider a scenario in which a user seeks to create a sparse variant of `SmoothGrad` that yields non-zero gradients for only a small number of features ("`SparseSmoothGrad`"). Designing `SparseSmoothGrad` only requires the addition of a regularization term to the loss function used in the `SmoothGrad` instance of the LFA framework (e.g. $\ell = \ell_{SmoothGrad} + \|\nabla_\xi g(\mathbf{x}_\xi)\|_0$), at which point, sparse solvers may be employed to solve the problem. Note that, unlike `SmoothGrad`, `SparseSmoothGrad` does not have a closed form solution, but that is not an issue for the LFA framework. More generally, by allowing for the customization of (1), (2), (3), and (4), the LFA framework creates new explanation methods through "variations on a theme".

We discuss the practical implications of Section §4 by providing the following recommendation for choosing among explanation methods.

Recommendation for choosing among explanation methods. In general, choose methods that satisfy the guiding principle of model recovery in the input domain in question. For continuous data, use additive continuous noise methods (e.g. `SmoothGrad`, `Vanilla Gradients`, `C-LIME`) or modified multiplicative continuous noise methods (e.g. `Integrated Gradients`, `Gradient x Input`) as described in §4.2. For binary data, use binary noise methods (e.g. `LIME`, `KernelSHAP`, `Occlusion`). Given the lack of methods for discrete noise, in case of discrete data, design novel explanation methods using the LFA framework with discrete noise neighbourhoods. Within each domain, choosing among appropriate methods boils down to determining the perturbation neighbourhood most suitable in the given context.

5 Empirical Evaluation

In this section, we present an empirical evaluation of the LFA framework. We first describe the experimental setup and then discuss three experiments and their findings.

5.1 Experimental Setup

Datasets. We experiment with two real-world datasets for two prediction tasks. The first dataset is the life expectancy dataset from the World Health Organization (WHO) [43]. It consists of countries' demographic, economic, and health factors, with 2,938 observations for 20 continuous features. We use this dataset to perform regression, predicting life expectancy. The other dataset is the home equity line of credit (HELOC) dataset from FICO [13]. It consists of information on HELOC applications, with 9,871 observations for 24 continuous features. We use this dataset to perform classification, predicting whether an applicant made payments without being 90 days overdue.

Models. For each dataset, we train four models: a simple model (linear regression for the WHO dataset and logistic regression for the HELOC dataset) that can satisfy conditions of the guiding principle and three more complex models (neural networks of varying complexity) that are more reflective of real-world applications. Model architecture and performance are described in Appendix A.4.

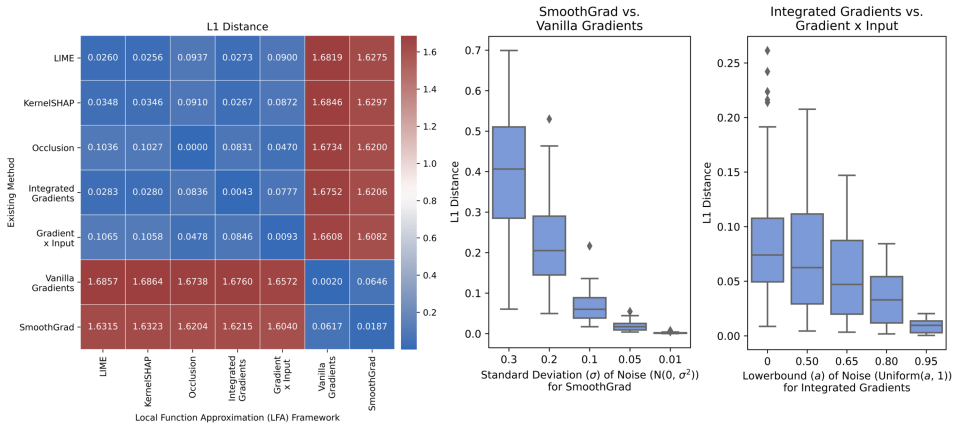
Metrics. To measure the similarity between two vectors (e.g. between two sets of explanations or between an explanation and the true model weights), we use L1 distance and cosine distance. L1 distance ranges between $[0, \infty)$ and is 0 when two vectors are the same. Cosine distance measures the angle between two vectors. It ranges between $[0, 2]$ and is 0 when the angle between two vectors is 0° (or 360°). For both metrics, the lower the value, the more similar two given vectors are.

5.2 Empirical Results and Findings

Here, we describe the experiments, present empirical findings, and discuss their implications.

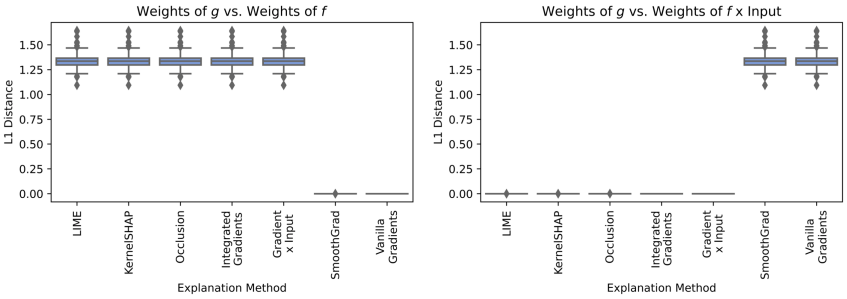
Existing explanation methods are instances of the LFA framework. First, we compare existing methods with corresponding instances of the LFA framework to assess whether they generate the same explanations. To this end, we use seven methods to explain the predictions of black-box models on 100 randomly-selected test set points. For each method, explanations are computed using either the existing method (implemented by Meta's open-source Captum library [22]) or the corresponding instance of the LFA framework (described in Table 1). The similarity of a given pair of explanations is measured using L1 distance or cosine distance.

The L1 distance values for a neural network with three hidden layers trained on the WHO dataset are shown in Figure 1. In Figure 1a, lowest L1 distance values appear in the diagonal of the heatmap, indicating that explanations generated by existing methods and corresponding instances of the LFA framework are very similar. Figures 1b and 1c show that explanations generated by instances of the LFA framework corresponding to `SmoothGrad` and `Integrated Gradients` converge to those of `Vanilla Gradients` and `Gradient x Input`, respectively. Together, these results demonstrate that, consistent with §3, existing methods are instances of the LFA framework. In addition, the clustering of the methods in Figure 1a indicates that, consistent with the analysis in § 4, for continuous data, `SmoothGrad` and `Vanilla Gradients` generate similar explanations while `LIME`, `KernelSHAP`,



(a) (b) (c)

Figure 1: Correspondence of existing methods and instances of the LFA framework. (**a**) Heatmap of average L1 distance between pairs of explanations. Boxplots of L1 distance between explanations of (**b**) `SmoothGrad` and `Vanilla Gradients` and (**c**) `Integrated Gradients` and `Gradient x Input`. The lower the L1 distance, the more similar two explanations are. Existing explanation methods are instances of the LFA framework.



(a) (b)

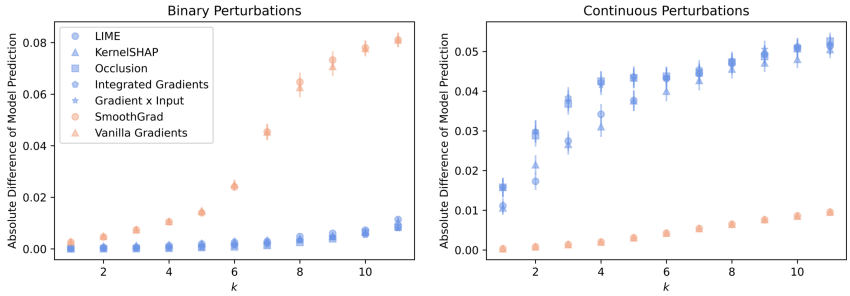
Figure 2: Analysis of model recovery. The lower the L1 distance, the more similar g's weights are to (**a**) f's weights or (**b**) f's weights multiplied by the input. For continuous data, additive continuous noise methods recover f's weights, satisfying the guiding principle, while multiplicative binary or continuous noise methods do not, recovering f's weights multiplied by the input instead.

`Occlusion`, `Integrated Gradients`, and `Gradient x Input` generate similar explanations. We observe similar results across various metrics, datasets, and models (Appendix A.5).

Some methods recover the underlying model while others do not (guiding principle). Next, we assess empirically which existing methods satisfy the guiding principle, i.e. which methods recover the black-box model f when f is of the interpretable model class $\mathcal{G}$. We specify a setting in which f and g are of the same model class, generate explanations using each method, and assess whether g recovers f for each explanation. For the WHO dataset, we set f and g to be linear regression models and generate explanations for 100 randomly-selected test set points. Then, for each point, we compare g's weights with f's gradients or with f's gradients multiplied by the input because, based on §4, some methods generate explanations on the scale of gradients while others on the scale of gradient-times-input. Note that, for linear regression, f's gradients are f's weights.

Results are shown in Figure 2. Consistent with §4, for continuous data, `SmoothGrad` and `Vanilla Gradients` recover the black-box model, thereby satisfying the guiding principle, while `LIME`, `KernelSHAP`, `Occlusion`, `Integrated Gradients`, and `Gradient x Input` do not. We observe similar results for the HELOC dataset using logistic regression models for f and g (Appendix A.5).

No single method performs best across all neighborhoods (no free lunch theorem). Lastly, we perform a set of experiments to illustrate the no free lunch theorem in § 4. We generate explanations for black-box model predictions for 100 randomly-selected test set points and evaluate the explanations using perturbation tests. The perturbation tests follow an intuition similar to that of saliency map evaluation in prior works [10, 37, 30]. For a given data point, k, and explanation, we identify the



(a) (b)

Figure 3: Perturbation tests perturbing bottom k features using (**a**) binary or (**b**) continuous noise. The lower the curve, the better a method identifies unimportant features. Results illustrate the no free lunch theorem: no single method performs best across all neighborhoods.

bottom-k features and either replace them with zero (binary perturbation) or add Gaussian noise to them (continuous perturbation). Then, we calculate the absolute difference in model prediction before and after perturbation. For each point, we generate one binary perturbation (since such perturbations are deterministic) and 100 continuous perturbations (since such perturbations are random), computing the average absolute difference in model prediction for the latter. In this setup, methods that better identify unimportant features yield smaller changes in model prediction.

Results of perturbation tests performed on explanations for a neural network with three hidden layers trained on the WHO dataset are displayed in Figure 3. Consistent with the no free lunch theorem, `LIME`, `KernelSHAP`, `Occlusion`, `Integrated Gradients`, and `Gradient x Input` perform best on binary perturbation neighborhoods (Figure 3a) while `SmoothGrad` and `Vanilla Gradients` perform best on continuous perturbation neighborhoods (Figure 3b). We observe similar results across metrics, datasets, and models (Appendix A.5). These findings have important implications: one should carefully consider the perturbation neighborhood not only when selecting a method to generate explanations but also when selecting a method to evaluate explanations. In fact, the type of perturbations used to evaluate explanations directly determines explanation method performance.

6 Conclusions and Future Work

In this work, we formalized the *local function approximation (LFA)* framework and demonstrated that various popular explanation methods can be characterized as instances of this framework with different notions of neighbourhood and different loss functions. We also introduced the no free lunch theorem, showing that no single method can perform optimally across all neighbourhoods, and provided a guiding principle for choosing among methods.

The function approximation perspective captures the essence of an explanation – a simplification of the real world (i.e. a black box model) that is nonetheless accurate enough to be useful (i.e. predict outcomes of a set of perturbations). When the real world is "simple", an explanation should completely capture its behaviour, a hallmark expressed precisely by the guiding principle.

Our work addresses key open questions in the field. In response to criticism about the lack of agreement in the field regarding the overall goals of post hoc explainability [26], our work points to function approximation as a principled goal. It also addresses the disagreement problem [23] and explains why different methods generate different explanations for the same model prediction. According to the LFA framework, this disagreement occurs because different methods approximate the black box model over different neighborhoods using different loss functions.

While our work makes several fundamental contributions, there is scope for further research along these lines. First, we analyzed seven popular post hoc explanation methods and this analysis could be extended to other methods. Second, our work focuses on the faithfulness (fidelity) rather than on the interpretability of explanations. The latter is encapsulated in the "interpretable" model class $\mathcal{G}$, which includes all the information about human preferences with regards to interpretability. However, it is unclear what precise notions of interpretability can be used to characterize such an interpretable model class. These are important directions for future research.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*. 9505–9515.

[2] Sushant Agarwal, Shahin Jabbari, Chirag Agarwal, Sohini Upadhyay, Steven Wu, and Himabindu Lakkaraju. 2021. Towards the unification and robustness of perturbation and gradient based explanations. In *International Conference on Machine Learning*. 110–119.

[3] David Alvarez-Melis and Tommi Jaakkola. 2018. On the Robustness of Interpretability Methods. *CoRR* abs/1806.08049 (2018).

[4] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. 2018. Towards better understanding of gradient-based attribution methods for Deep Neural Networks. In *International Conference on Learning Representations*.

[5] Solon Barocas, Andrew Selbst, and Manish Raghavan. 2020. The hidden assumptions behind counterfactual explanations and principal reasons. In *ACM Conference on Fairness, Accountability, and Transparency*. 80–89.

[6] Osbert Bastani, Carolyn Kim, and Hamsa Bastani. 2017. Interpretability via model extraction. *CoRR, abs/1706.09773* (2017).

[7] Prasad Chalasani, Jiefeng Chen, Amrita Roy Chowdhury, Xi Wu, and Somesh Jha. 2020. Concise Explanations of Neural Networks using Adversarial Training. In *International Conference on Machine Learning*. 1383–1391.

[8] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. 2018. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 839–847.

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

[10] Piotr Dabkowski and Yarin Gal. 2017. Real time image saliency for black box classifiers. *Advances in Neural Information Processing Systems* 30 (2017).

[11] Ann-Kathrin Dombrowski, Maximilian Alber, Christopher Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. 2019. Explanations can be manipulated and geometry is to blame. *CoRR, abs/1906.07983* (2019).

[12] Radwa Elshawi, Mouaz H Al-Mallah, and Sherif Sakr. 2019. On the interpretability of machine learning-based model for predicting hypertension. *BMC medical informatics and decision making* 19, 1 (2019), 146.

[13] FICO. 2019. Home equity line of credit (HELOC) dataset. *Explainable Machine Learning Challenge* (2019).

[14] Damien Garreau and Ulrike von Luxburg. 2020. Looking deeper into LIME. *CoRR* abs/2008.11092 (2020).

[15] Amirata Ghorbani, Abubakar Abid, and James Zou. 2019. Interpretation of neural networks is fragile. In *AAAI Conference on Artificial Intelligence*, Vol. 33. 3681–3688.

[16] Aapo Hyvärinen. 2005. Estimation of Non-Normalized Statistical Models by Score Matching. *Journal of Machine Learning Research* 6, 24 (2005), 695–709. http://jmlr.org/papers/v6/hyvarinen05a.html

[17] Mark Ibrahim, Melissa Louie, Ceena Modarres, and John Paisley. 2019. Global Explanations of Neural Networks: Mapping the Landscape of Predictions. *CoRR, abs/1902.02384* (2019).

[18] Amir-Hossein Karimi, Gilles Barthe, Borja Balle, and Isabel Valera. 2019. Model-Agnostic Counterfactual Explanations for Consequential Decisions. arXiv:1905.11190 [cs.LG]

[19] Amir-Hossein Karimi, Bernhard Schölkopf, and Isabel Valera. 2020. Algorithmic recourse: from counterfactual explanations to interventions. *CoRR, abs/2002.06278* (2020).

[20] Amir-Hossein Karimi, Julius von Kügelgen, Bernhard Schölkopf, and Isabel Valera. 2020. Algorithmic recourse under imperfect causal knowledge: a probabilistic approach. *CoRR, abs/2006.06831* (2020).

[21] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie J. Cai, James Wexler, Fernanda B. Viégas, and Rory Sayres. 2018. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *International Conference on Machine Learning*.

[22] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A unified and generic model interpretability library for PyTorch. arXiv:2009.07896 [cs.LG]

[23] Satyapriya Krishna*, Tessa Han*, Alex Gu, Javin Pombra, Shahin Jabbari, Steven Wu, and Himabindu Lakkaraju. 2022. The Disagreement Problem in Explainable Machine Learning: A Practitioner's Perspective. *arXiv preprint arXiv:2202.01602* (2022).

[24] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. 2019. Faithful and Customizable Explanations of Black Box Models. In *AAAI Conference on Artificial Intelligence, Ethics, and Society*. 131–138.

[25] Alexander Levine, Sahil Singla, and Soheil Feizi. 2019. Certifiably robust interpretation in deep learning. *CoRR, abs/1905.12105* (2019).

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

[27] Arnaud Looveren and Janis Klaise. 2019. Interpretable Counterfactual Explanations Guided by Prototypes. *CoRR, abs/ 1907.02584* (2019).

[28] Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems* 30 (2017).

[29] Weili Nie, Yang Zhang, and Ankit Patel. 2018. A theoretical explanation for perplexing behaviors of backpropagation-based visualizations. In *International Conference on Machine Learning*. PMLR, 3809–3818.

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

[31] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. 2020. FACE: Feasible and Actionable Counterfactual Explanations. In *AAAI/ACM Conference on AI, Ethics, and Society*. 344–350.

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

[33] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*. 618–626.

[34] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. 2020. How can we fool LIME and SHAP? Adversarial Attacks on Post hoc Explanation Methods. In *AAAI/ACM Conference on AI, Ethics, and Society*. 180–186.

[35] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. SmoothGrad: Removing noise by adding noise. *arXiv preprint arXiv:1706.03825* (2017).

[36] Suraj Srinivas and François Fleuret. 2018. Knowledge transfer with Jacobian matching. In *International Conference on Machine Learning*. 4723–4731.

[37] Suraj Srinivas and François Fleuret. 2019. Full-gradient representation for neural network visualization. *Advances in neural information processing systems* 32 (2019).

[38] Suraj Srinivas and Francois Fleuret. 2021. Rethinking the Role of Gradient-based Attribution Methods for Model Interpretability. In *International Conference on Learning Representations*. https://openreview.net/forum?id=dYeAHXnpWJ4

[39] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*. 3319–3328.

[40] Berk Ustun, Alexander Spangher, and Yang Liu. 2019. Actionable recourse in linear classification. In *ACM Conference on Fairness, Accountability, and Transparency*. 10–19.

[41] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. *Harvard Journal of Law & Technology* 31 (2017), 841.

[42] Leanne S Whitmore, Anthe George, and Corey M Hudson. 2016. Mapping chemical performance on molecular structures using locally interpretable explanations. *CoRR, abs/1611.07443* (2016).

[43] World Health Organization (WHO). 2018. Life expectancy dataset. *Global Health Observatory Data Repository* (2018).

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Abstract, Section §1.
   (b) Did you describe the limitations of your work? [Yes] See Section §6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section §6.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Sections §3, §4, and Appendix.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Sections §3, §4, and Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We plan to include a zip file with the code in the supplementary materials.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Figure 3 and Appendix.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 5.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We do not directly obtain data from individuals.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] To our knowledge, the data contains no such information nor content. See Appendix.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We did not use crowdsourcing nor did we conduct research with human subjects.

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] We did not use crowdsourcing nor did we conduct research with human subjects.

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] We did not use crowdsourcing nor did we conduct research with human subjects.