CONSISTENT AND TRUTHFUL INTERPRETATION WITH FOURIER ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

For many interdisciplinary fields, ML interpretations need to be consistent with *what-if* scenarios related to the current case, i.e., if one factor changes, how does the model react? Although the attribution methods are supported by the elegant axiomatic systems, they mainly focus on individual inputs, and are generally inconsistent. To support what-if scenarios, we introduce a new objective of consistency based on a notion called truthful interpretation. Towards this objective, we apply Fourier analysis of Boolean functions to get consistency guarantees. Experimental results show that for neighborhoods with various radii, our method achieves 2x - 50x lower inconsistency compared with the other methods.

1 INTRODUCTION

Interpretability is a central problem in deep learning. During training, the neural network strives to minimize the training loss without other distracting objectives. However, to interpret the network, we have to construct a different model[1], which tends to have simpler structures and fewer parameters, e.g., a decision tree or a polynomial. Theoretically, these restricted models cannot perfectly interpret deep networks due to their limited representation power. Therefore, the previous researchers had to introduce various relaxations. The most popular and elegant direction is the attribution methods with axiomatic systems (Sundararajan et al., 2017; Lundberg & Lee, 2017), which mainly focus on individual inputs.

The interpretations of the attribution methods do not automatically extend to the neighboring points. Take SHAP (Lundberg & Lee, 2017) as the motivating example, illustrated in Figure 1 on the task of sentiment analysis of movie reviews. In this example, the interpretations of the two slightly different sentences are not consistent. It is not only because the weights of each word are significantly different but also because, after removing a word "very" of weight 19.9%, the network's output only drops by $97.8\% - 88.7\% = 9.1\%$. In other words, **the interpretation does not explain the network's behavior even in a small neighborhood of the input**.

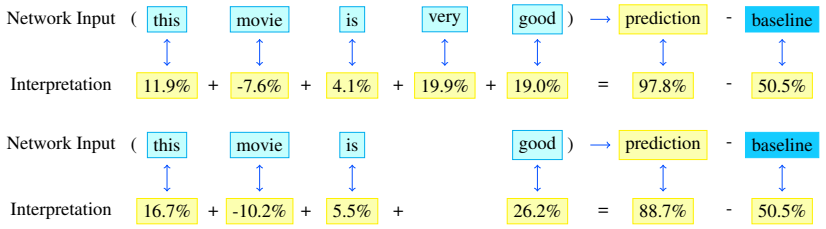


Figure 1: Interpretations generated by SHAP on movie review

Inconsistency is not a vacuous concern. Imagine a doctor treating a diabetic with the help of an AI system. The patient has features A, B, and C, representing three positive signals from various tests. AI recommends giving 4 units of insulin with the following explanation: A, B, and C have weights 1, 1, and 2, respectively, so 4 units in total. Then the doctor may ask AI: *what if* the patient only

[1]For simplicity, below we use **model** to denote the model that provides interpretation and **network** to denote the general black-box machine learning model that needs interpretation.

has A and B, but not C? One may expect the answer to be close to 2, as A+B has a weight of 2. However, the network is highly non-linear and may output other suggestions like 3 units, explaining that both A and B have a weight of 1.5. Such inconsistent behaviors will drastically reduce the doctor's confidence in the interpretations, limiting the AI system's practical value[2].

Unfortunately, the three elements – interpretability, consistency and efficiency – **form an impossible trinity**. Efficiency is a commonly used axiom in the attribution methods (Weber, 1988; Friedman & Moulin, 1999; Sundararajan & Najmi, 2020), also called local accuracy (Lundberg & Lee, 2017) or completeness (Sundararajan et al., 2017), which states that the model's output should be equal to the network's output for the given input. To ensure consistency, we have to use *the same function* g for interpreting the neighboring predictions of the network f for the input x. However, as we have discussed previously, the representation power of g is much smaller than f, so g cannot perfectly fit f even in a small neighborhood of x. Therefore, efficiency cannot be achieved with the same g function. In summary,

(a) Attribution methods are interpretable and efficient, but not consistent.

(b) The original network is consistent and efficient, but not interpretable.

(c) If one model is interpretable and consistent, it cannot be efficient.

Taking one step back, efficiency is not a necessity in interpretability. For example, when interpreting one person's behavior, there always exists uncertainties even with abundant evidence and background knowledge. Interestingly, the fact that the model g being weaker does not mean it has to be wrong. Indeed, g can still truthfully represent the "readable part" of f (see Definition 3.5). The other part of f, i.e., the "unreadable part", is not simply a "fitting error". Instead, it truthfully represents the higher order non-linearities in the network that our interpretation model g, even doing its best, cannot cover. In short, **what g tells is true, although it does not cover all the truth.**

Revisiting the healthcare example, when the patient has features A, B and C, and AI gives 4 units of insulin, we may provide the following interpretation: A, B, and C have weights 1, 1, and 1.5, respectively, and there exists an uncertainty of 0.5 (see Definition 3.7), which is due to the correlations among A, B, C. When the feature C disappears, AI gives 3 units of insulin, and our interpretation becomes: the weights of A and B do not change, but this time we have a larger uncertainty of 1, which add up to 3 units. Here, the interpretation is *truthful*, in the sense that it accurately describes the linear contributions of the three features, and the uncertainties only contain higher order factors. Such combination has great clinical value, as the doctors clearly know which part can be exactly interpreted and which cannot.

Uncertainty naturally leads to the notion of inconsistency:

Definition 1.1 (Inconsistency)**.** *given an input $x \in \mathcal{X}$ and functions $f, g \in \mathcal{X} \to \mathcal{Y}$, the inconsistency between f and g with respect to the weight distribution $\mathcal{D}$ is*

$$\mathbb{I}_{\mathcal{D}}(f, g) = \mathbb{E}_{x \in \mathcal{D}} \|f(x) - g(x)\|_2^2 \tag{1}$$

Here $\mathcal{D}$ is usually defined as a uniform distribution on the neighborhood of input x (under a certain metric), in which case we denote the distribution as $\mathcal{N}_x$. With this definition, there is a clear trade-off between inconsistency and the "radius" of the neighborhood. Specifically, if $\mathcal{N}_x$ only contains x, even constant g gives 0-inconsistency; if $\mathcal{N}_x = \mathcal{X}$, the inconsistency is inevitably much worse. Therefore, we want to find a truthful g that minimizes the inconsistency while maximizing the neighborhood.

In this paper, we focus on the case that f and g are Boolean functions, i.e., the input variables are binary. Previous works based on Shapley value also assume binary inputs, see e.g. (Lundberg & Lee, 2017; Sundararajan et al., 2020). Given any Boolean function, we may expand it on the Fourier basis and get its Fourier spectrum. It is well known that the Fourier spectrum's complexity naturally characterizes the complexity of Boolean functions (O'Donnell, 2014), e.g., a small decision tree can be approximated with a low degree sparse polynomial on the Fourier basis (Mansour, 1994). Therefore, we restrict g functions on its Fourier spectrum. By applying the Harmonica algorithm (Hazan et al., 2017), we can achieve theoretically guaranteed low inconsistency using g for the target f, when f is "approximately sparse" (see Definition 3.8).

[2]We are not the first to note the importance of consistency. Previously, Ribeiro et al. (2016) have investigated it under the name *local fidelity*. However, they did not provide rigorous theoretical guarantees for consistency.

After formally introducing Harmonica and its theoretical guarantees in Section 4, we will compare it with the existing methods in Section 5. Afterwards, In Section 6.2, we will show that on datasets like SST-2 and IMDb, Harmonica can get 2x-50x lower inconsistency compared with other methods.

2 Related Work

Interpretability is a critical topic in machine learning, and we refer the reader to (Doran et al., 2017; Lipton, 2018) for insightful general discussions. Below we discuss different types of interpretable models.

Model-specific interpretable models Interpretable/white-box models are inherently ante-hoc and model-specific. One of the goals behind using interpretable models is to have inherent model interpretability. Current mainstream approaches include: Decision Trees (Wang et al. (2015a), Balestriero (2017), Yang et al. (2018)), Decision Rules (Wang et al. (2015b), Su et al. (2015)), Decision Sets (Lakkaraju et al. (2019), Wang et al. (2017)) and Linear Model (Ustun & Rudin (2014), Ustun et al. (2013)).

Shapley value based explanations Shapley value (Shapley, 1953) was firstly introduced in cooperative game theory, with several strong axiomatic theoreticlal properties (Weber, 1988; Grabisch & Roubens, 1999). Recently, it has been adopted for explanations of machine learning models (Lundberg & Lee, 2017; Štrumbelj & Kononenko, 2014; Sundararajan & Najmi, 2020; Wang et al., 2021a; Zhang et al., 2021; Frye et al., 2020; Yuan et al., 2021) or feature importance (Covert et al., 2020).

Based on the Shapley value, Owen (1972) proposed the Shapley interaction value to study pairwise interactions between players. Grabisch & Roubens (1999) generalized it to study interactions of higher orders and provided an axiomatic foundation. Starting from that, many researchers worked on higher-order feature interactions from different perspectives (Sundararajan et al., 2020; Masoomi et al., 2022; Tsang et al., 2020; Aas et al., 2021).

However, computing Shapley value is NP-hard (Elkind et al., 2009; Van den Broeck et al., 2021). Recently many works on Shapley value focus on how to approximate Shapley value efficiently, including sampling-based methods, regression-based methods, or learning-based methods (Lundberg & Lee, 2017; Chen et al., 2018; Ancona et al., 2019; Covert & Lee, 2021; Jethani et al., 2021; Hamilton et al., 2021; Wang et al., 2021b).

Gradient-based explanations Simonyan et al. (2013) proposed a method to approximate the network and propose the gradient as attribution linearly. Smoothgrad (Smilkov et al., 2017) generates explanations by introducing perturbation to the input and then observing the corresponding effect on the model's predictions. Integrated Gradients (Sundararajan et al., 2017) distributes the change in output with respect to a baseline input by integrating gradients between the two input states, and Janizek et al. (2021) generalized this method to second order.

DeepLIFT (Shrikumar et al., 2017) and LRP (Montavon et al., 2017) backpropagated contribution layer-wise to every feature of the input. Grad-CAM (Selvaraju et al., 2017) leverage activation values of convolutional layers. CXPlain (Schwab & Karlen, 2019) removes a single feature from a single input and measures the change in the loss function.

Other methods (Covert et al., 2021) introduced a general removal-based framework that unifies 26 interpretability algorithms, including classical methods like LIME (Ribeiro et al., 2016), DeepLIFT (Shrikumar et al., 2017), LRP (Montavon et al., 2017), etc.

3 Preliminaries

In this paper, we assume there are n input variables, and all of them are binary, so the input space $\mathcal{X}$ is $\{-1, 1\}^n$. We use $-1/1$ instead of $0/1$ to represent the binary variables, because it easily fits into the Fourier basis. We assume the output is a real number, so the output space $\mathcal{Y}$ is $\mathbb{R}$, but generally the output can be anything (e.g., a vector representing the classification outcome).

3.1 FOURIER ANALYSIS

Fourier analysis of Boolean function is a fascinating field, and we refer the reader to O'Donnell (2014) for more comprehensive materials. Here we first introduce the Fourier basis:

Definition 3.1 (Fourier basis)**.** *For any subset of variables $S \subseteq [n]$, we define the corresponding Fourier basis as*

$$\chi_S = \Pi_{i \in S} x_i \ \in \{-1,1\}^n \to \{-1,1\}$$

The Fourier basis is a complete orthonormal basis for Boolean functions, under the uniform distribution on $\{-1,1\}^n$. We define the inner product as follows.

Definition 3.2 (Inner product)**.** *Given two functions $f, g \in \{-1,1\}^n \to \mathbb{R}$, their inner product is:*

$$\langle f, g \rangle = 2^{-n} \sum_{x \in \{-1,1\}^n} f(x)g(x) = \mathop{\mathbb{E}}_{x \sim \{-1,1\}^n} [f(x)g(x)]$$

Then we can compute the Fourier spectrum of any Boolean functions based on the Fourier basis.

Definition 3.3 (Fourier expansion)**.** *For any Boolean function $f \in \{-1,1\}^n \to \mathbb{R}$, we can expand it as*

$$f(x) = \sum_{S \subseteq [n]} \hat{f}_S \chi_S(x),$$

where $\hat{f}_S = \langle f, \chi_S \rangle$ is the Fourier coefficient on S. All the Fourier coefficients together are called the Fourier spectrum of f.

Inner product defines one kind of similarity of two functions, and is invariant under different basis. Specifically, we have the following Theorem.

Definition 3.4 (Plancherel's Theorem)**.** *Given two functions $f, g \in \{0,1\}^n \to \mathbb{R}$,*

$$\langle f, g \rangle = \sum_{S \subseteq [n]} \hat{f}(S)\hat{g}(S)$$

When setting $f = g$, we get the Parseval's identity: $\mathbb{E}[f^2] = \sum_S \hat{f}_S^2$.

3.2 INTERPRETABILITY

Based on the notions from Fourier analysis, we first define the set of human readable functions.

Definition 3.5 (C-Readable function)**.** *Given a set of bases C, a function f is C-readable if it is supported on C. That is, for any $\chi_S \notin C$, $\langle f, \chi_S \rangle = 0$.*

Readable notion is parameterized with C, because it may differ case by case. If we set C to be all the single variable bases, only linear functions are readable; if we set C to be all the bases with degree at most 2, functions with pairwise interactions are also readable. Moreover, if we further add one higher order term to C, e.g., $\chi_{\{x_1,x_2,x_3,x_4\}}$, it means we can also reason about the factor $x_1x_2x_3x_4$ in the interpretation, which might be an important empirical factor that people can easily understand. When a function faithfully interprets the readable part of the network, it is called a truthful interpretation. Such faithfulness is characterized using the following notion.

Definition 3.6 (Readable inconsistency)**.** *Given a set of bases C, two functions $f, g \in \{0,1\}^n \to \mathbb{R}$, the readable inconsistency of g to f with respect to C is defined as*

$$\mathbb{R}_C(f, g) = \sum_{\chi_S \in C} \langle f - g, \chi_S \rangle^2 \tag{2}$$

Due to Parseval's identity, given a C-readable function g, the inconsistency defined in Eqn. (1) under the uniform distribution of the whole space, can be decomposed into two parts:

$$\mathbb{I}_{\mathcal{D}}(f, g) = \sum_{\chi_S, S \subseteq [n]} \langle f - g, \chi_S \rangle^2 = \mathbb{R}_C(f, g) + \sum_{\chi_S \notin C} \langle f - g, \chi_S \rangle^2 = \mathbb{R}_C(f, g) + \sum_{\chi_S \notin C} \langle f, \chi_S \rangle^2$$

Where the last equality holds due to the definition of C-readable. When $\mathbb{R}_C(f, g)$ is small, $f - g$ has negligible weights on the bases in C, so f and g are almost consistent on C.

Inconsistency is defined over the whole function space, and below we also define the pointwise uncertainty.

Definition 3.7 (Uncertainty)**.** *Given two functions $f, g \in \{0,1\}^n \to \mathbb{R}$, an input x, the uncertainty of the interpretation of g to f on x is $u(x) = |f(x) - g(x)|$.*

To present the theoretical guarantees of the Harmonica algorithm, we introduce the following definition, which is slightly different from its original version in Hazan et al. (2017).

Definition 3.8 (Approximately sparse function)**.** *We say a function $f \in \{-1,1\}^n \to \mathbb{R}$ is (ϵ, s, C)-bounded, if $\mathbb{E}[(f - \sum_{\chi_S \in C} \hat{f}(S)\chi_S)^2] \geq 1 - \epsilon$ and $\sum_S |\hat{f}(S)| \leq s$.*

Here f is (ϵ, s, C)-bounded means it is almost readable, and has bounded ℓ_1 norm.

4 Learning Boolean functions

With the previous definitions, it becomes clear that finding a truthful interpretation g is equivalent to accurately learning a Boolean function with respect to the readable bases set C. Learning Boolean function is a classical problem in learning theory, and there are many algorithms like KM algorithm (Kushilevitz & Mansour, 1991), Low-degree algorithm (Linial et al., 1993) and Harmonica (Hazan et al., 2017).

However, there are two main differences between our setting and the previous ones:

- The previous algorithms assume a black-box oracle of the Boolean function, and the main objective is to learn the function and minimize the oracle query complexity. In our setting, we also have an oracle, i.e., the network function f, but the oracle queries are not expensive. Moreover, we are not learning the whole f, but only its readable part instead.
- The previous algorithms assume the target function f is "sparse", or more specifically, with small ℓ_1 norm in the Fourier basis. In our setting, this is not necessarily true. Our network f might be extremely complicated, but we can still provide truthful readable interpretation, which is sparse due to the fact that the bases set C is usually small.

Based on these observations, we pick two algorithms for our task: Harmonica and Low-degree. Compared with Low-degree, Harmonica has much better sampling efficiency based on the compressed sensing techniques. Specifically, for general real-valued decision trees which has s leaf nodes and are bounded by B, the sample complexity of Harmonica is $\tilde{O}\left(B^2 s^2/\varepsilon \cdot \log n\right)$, while Low-degree is $\tilde{O}\left(B^4 s^2/\varepsilon^2 \cdot \log n\right)$ (Hazan et al., 2017). However, the theoretical guarantee of Harmonica depends on the assumption that f is approximately sparse, which is not necessarily true for deep networks. Empirically we observe that Harmonica work well for various cases, but we remark that Low-degree does not depend on sparsity of f, so it is theoretically more general.

4.1 Harmonica algorithm

Algorithm 1 Harmonica

1. Given uniformly randomly sampled $x_1, \cdots, x_T$, evaluate them on f: $\{f(x_1), ..., f(x_T)\}$.
2. Solve the following regularized regression problem.

$$\underset{\alpha \in \mathbb{R}^{|C|}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{T} \left(\sum_{S, \chi_S \in C} \alpha_S \chi_S(x_i) - f(x_i) \right)^2 + \lambda \|\alpha\|_1 \right\} \tag{3}$$

3. Output the polynomial $g(x) = \sum_{S, \chi_S \in C} \alpha_S \chi_S(x)$.

Our algorithm is slightly different from the original algorithm proposed by Hazan et al. (2017), but similar theoretical guarantees still hold. We include the proof in the Appendix A.

Theorem 4.1. *Given $f \in \{-1,1\}^n \to \mathbb{R}$ a $(\epsilon/4, s, C)$-bounded function, Algorithm 1 finds a function g with inconsistency at most ϵ in time $O((T \log \frac{1}{\epsilon} + |C|/\epsilon) \cdot |C|)$ and sample complexity $T = \tilde{O}(s^2/\epsilon \cdot \log |C|)$.*

4.2 Low-degree algorithm

Low-degree algorithm is based on the concentration inequality, and it estimates the coefficient of each axis individually.

Algorithm 2 Low-degree

1. Given uniformly randomly sampled $x_1, \cdots, x_T$, evaluate them on f: $\{f(x_1), ..., f(x_T)\}$.
2. For any $\chi_S \in C$, let $\alpha_S = \frac{\sum_{i=1}^{T} f(x_i)\chi_S(x_i)}{T}$.
3. Output the polynomial $g(x) = \sum_{S, \chi_S \in C} \alpha_S \chi_S(x)$.

Theorem 4.2 (Linial et al. (1993))**.** *Given any $\epsilon, \delta > 0$, assuming that function is bounded by B, when $T \geq \frac{2B^2}{\epsilon^2} \log \frac{2|C|}{\delta}$, we have*

$$\Pr\left[\forall \chi_S \in C, s.t., |\hat{g}(S) - \hat{f}(s)| \leq \epsilon\right] \geq 1 - \delta$$

Theorem 4.2 was proved using Hoeffding bound, and we included the proof in Appendix B for completeness. Although Low-degree algorithm does not need strong assumptions for the target function f, empirically we observe Harmonica works much more efficiently. So in our experiments, we use Harmonica instead.

5 Discussion on the existing algorithms

In this section, we compare our approach with the existing techniques from the perspective of consistency and truthfulness, and then manually investigate every method's performance on a simple three-parameter majority function.

LIME (Ribeiro et al., 2016) Given an input x, Lime samples the neighborhood points based on a sampling distribution Π_x, and optimizes the following program:

$$\min_{g \in \mathcal{G}} L(f, g, \Pi_x) + \Omega(g)$$

where L is the loss function describing the distance between f and g on the sampled data points, $\mathcal{G}$ is the set of readable functions (e.g. the set of linear functions), $\Omega(\cdot)$ is a function that characterizes the complexity of g. In other words, LIME tries to minimize the fitting error and simultaneously minimizes the complexity of g (which is usually the sparsity of the linear function). By minimizing L, LIME also works towards better consistency, but their approach is purely heuristic, without any theoretical guarantees. Although their readable function set can easily generalize to the set with higher order terms, the sampling distribution Π_x is not uniform, so it is difficult to incorporate the orthonormal basis into their framework. In other words, the model they compute is not truthful.

Attribution methods As we discussed in introduction, attribution methods mainly focus on individual inputs, instead of neighboring points. Therefore, it is difficult for attribution methods to achieve low inconsistency, especially for first order methods like SHAP (Lundberg & Lee, 2017) and IG (Sundararajan et al., 2017).

For higher order attribution methods, consistency can potentially be improved due to their enhanced representation power. The classical Shapley interaction index has the problem of not fitting the underlying function therefore not efficient, as observed by Sundararajan et al. (2020), who proposed Shapley Taylor interaction index (Sundararajan et al., 2020) with better empirical performance. Shapley Taylor interaction index satisfies a generalized efficiency axiom, which says for all $f \in \{-1,1\}^n \to \mathbb{R}$,

$$\sum_{S \subseteq [n], |S| \leq k} \mathcal{I}_S^k(f) = f([n]) - f(\emptyset)$$

We should remark that both Shapley interaction index and Shapley Taylor interaction index were not originally designed for consistent interpretations, so they did not specify how to generalize the interpretation for neighboring points. To this end, we make a global extension to the Shapley value based interpretation, that is, using Shapley interaction indices or Shapley Taylor interaction indices as the coefficients of corresponding terms of the polynomial surrogate function.

$$g\left(x_1, x_2, \cdots, x_n\right) = f(\emptyset) + \sum_{x_i \in S, S \subseteq N} \mathcal{I}(f, S)$$

However, these higher order Shapley value based methods all focus on the original Shapley value framework, so their interpretations are not truthful, i.e., not getting the exact coefficient of f even on the "simple bases". Moreover, as we will show in our experiments, higher order methods still incur high inconsistency compared with our methods.

When applying Shapley value techniques for visual search, Hamilton et al. (2021) proposed an interesting and novel sampling + Lasso regression algorithm for efficiently computing higher order Shapley Taylor index in their experiments. However, their methods are based on a sampling probability distribution generated from permutation numbers, which is far from the uniform distribution. Additionally, their algorithm is based on the Shapley Taylor index, so their method is not truthful as well.

5.1 Test with the Majority Function

To investigate the consistency of different interpretation methods, let us take a closer look at the majority function $\mathrm{Maj}_3 : \{-1, 1\}^3 \to \{-1, 1\}$, which is defined as $\mathrm{Maj}_n(x) = \mathrm{sgn}\left(x_1 + x_2 + x_3\right)$. It is easy to verify the Fourier expansion for Maj_3 is:

$$\mathrm{Maj}_3\left(x_1, x_2, x_3\right) = \frac{1}{2}x_1 + \frac{1}{2}x_2 + \frac{1}{2}x_3 - \frac{1}{2}x_1x_2x_3$$

For this simple function, we can manually compute the outcome of each algorithm, as illustrate in Table 1. If the output of the algorithm is correct, i.e., is equal to the output of Maj_3, we write a check mark. Otherwise, we write down the actual difference between the two outputs.

As we can see, we are the only method that is consistent for the whole function values. Third order Shapley Taylor index is also very good, only missing three inputs. Other methods can only fit a few inputs, and LIME misses all the cases because Maj_3 is not a linear function.

Algorithms	$(-1,-1,-1)$	$(-1,-1,+1)$	$(-1,+1,-1)$	$(-1,+1,+1)$	$(+1,-1,-1)$	$(+1,-1,+1)$	$(+1,+1,-1)$	$(+1,+1,+1)$
Harmonica (3-order)	✔	✔	✔	✔	✔	✔	✔	✔
LIME	-1.5	-0.50	-0.50	$+0.50$	-0.50	$+0.50$	$+0.50$	$+1.5$
SHAP	✔	-0.33	-0.33	$+0.33$	-0.33	$+0.33$	$+0.33$	✔
Shapley Interaction Index	✔	-0.33	-0.33	$+0.33$	-0.33	$+0.33$	$+0.33$	-3.0
Shapley Taylor Index (3-order)	✔	✔	✔	-0.33	✔	-0.33	-0.33	✔

Table 1: Consistency of Interpretations by Harmonica, LIME, SHAP, Shapley Interaction Index, and Shapley Taylor Interaction Index on Majority function Maj_3.

6 Experiments

In this section, we conduct experiments on NLP tasks to evaluate the inconsistency $\mathbb{I}_{\mathcal{D}}(f, g)$ and readable inconsistency $\mathbb{R}_C(f, g)$ of different algorithms quantitatively. In our experiment, we choose 2-order and 3-order Harmonica algorithms, which correspond to setting C to be all terms with the order at most 2 and 3. We did not include special higher terms in our experiment because without other prior knowledge, lower-order terms are easier to understand for human beings.

The baseline algorithms chosen for comparison include LIME (Ribeiro et al., 2016), Integrated Gradients (Sundararajan et al., 2017), SHAP (Lundberg & Lee, 2017), Integrated Hessians (Janizek et al., 2021), and Shapley Taylor interaction index (Sundararajan et al., 2020; Hamilton et al., 2021), where the first three are first-order algorithms, and the last two are second-order algorithms.

The two NLP tasks we pick are the SST-2 (Socher et al., 2013) dataset for sentiment analysis and the IMDb (Maas et al., 2011) dataset for movie review classification.

6.1 Language Interpretation

Starting with a binary sentiment classification task on SST-2 dataset, we try to interpret a convolutional neural network trained with Adam (Kingma & Ba, 2014) optimizer for 10 epochs. The model has a test accuracy of 80.61%. For a given input sentence x with length l, We define the induced neighborhood $\mathcal{N}_x$ by introducing a masking operation on this sentence. The radius $0 \leq r \leq l$ is defined as the maximum number of masked words.

Estimating Inconsistency Since exactly calculating the inconsistency by definition in Eqn.(1) is expensive, we sample multiple $x \in \mathcal{D}$ to estimate it. By concentration inequality, the estimation error shrinks exponentially fast as the sample size goes up: $\mathrm{P}\left(\left|\hat{\mathbb{I}_{\mathcal{D}}}(f,g) - \mathbb{I}_{\mathcal{D}}(f,g)\right| \geq \frac{t}{s}\right) \leq 2\exp\left(-\frac{2t^2}{s}\right)$. In our experiments, we choose the sample size s to be 2×10^6, hence we have $\mathrm{P}\left(\left|\hat{\mathbb{I}_{\mathcal{D}}}(f,g) - \mathbb{I}_{\mathcal{D}}(f,g)\right| \geq 8.95 \times 10^{-4}\right) \leq 0.05$.

Results on Inconsistency Figure 2 and Table 2 show the inconsistency evaluated under different neighborhood with radius ranging from 1 to ∞ (for notation convenience, in real-world scenarios, the length of sentence l is limited). The inconsistency is evaluated under L^2, L^1, and L^0 norm (for L^0 norm, we set the threshold to be 0.1).

Here L^1 and L^0 norm inconsistencies are defined similar to Eqn.(1), replacing $\|f(x) - g(x)\|_2^2$ with $\|f(x) - g(x)\|_2$ and $\mathbb{1}\{\|f(x) - g(x)\|_2 \geq 0.1\}$. We can see that Harmonica consistently outperforms all the other baselines on all radii.

radius	L^2 norm	L^1 norm	L^0 norm
1	0.0006	0.0141	0.0073
2	0.0017	0.0207	0.0318
4	0.0037	0.0331	0.0830
8	0.0068	0.0502	0.1562
16	0.0100	0.0647	0.2170
32	0.0107	0.0674	0.2273
∞	0.0107	0.0674	0.2279

Harmonica^2

radius	L^2 norm	L^1 norm	L^0 norm
1	**0.0004**	0.0107	0.0067
2	**0.0014**	0.0179	0.0257
4	**0.0032**	0.0301	0.0707
8	**0.0060**	0.0455	0.1382
16	**0.0087**	0.0580	0.1925
32	**0.0093**	0.0602	0.2014
∞	**0.0093**	0.0600	0.2010

Harmonica^3

radius	L^2 norm	L^1 norm	L^0 norm
1	0.2313	0.4019	0.8503
2	0.2227	0.3817	0.8036
4	0.1968	0.3428	0.7449
8	0.1441	0.2811	0.6857
16	0.0985	0.2303	0.6460
32	0.0932	0.2243	0.6414
∞	0.0933	0.2246	0.6414

Harmonica-local^2

radius	L^2 norm	L^1 norm	L^0 norm
1	0.1117	0.2690	0.7647
2	0.1073	0.2622	0.7528
4	0.0982	0.2478	0.7232
8	0.0838	0.2254	0.6798
16	0.0721	0.2060	0.6433
32	0.0706	0.2035	0.6392
∞	0.0707	0.2035	0.6388

Harmonica-local^3

radius	L^2 norm	L^1 norm	L^0 norm
1	0.0125	0.0866	0.3538
2	0.0127	0.0859	0.3444
4	0.0132	0.0864	0.3418
8	0.0153	0.0921	0.3666
16	0.0188	0.1017	0.4035
32	0.0197	0.1040	0.4113
∞	0.0196	0.1039	0.4110

LIME

radius	L^2 norm	L^1 norm	L^0 norm
1	0.0149	0.0968	0.3539
2	0.0205	0.1119	0.4437
4	0.0279	0.1318	0.5219
8	0.0341	0.1475	0.5716
16	0.0365	0.1532	0.5873
32	0.0369	0.1540	0.5893
∞	0.0369	0.1540	0.5893

SHAP

radius	L^2 norm	L^1 norm	L^0 norm
1	0.0169	0.0682	0.2006
2	0.0308	0.1055	0.3351
4	0.0542	0.1566	0.5072
8	0.0819	0.2077	0.6474
16	0.0970	0.2337	0.7057
32	0.0981	0.2359	0.7104
∞	0.0979	0.2356	0.7103

Integrated Gradients

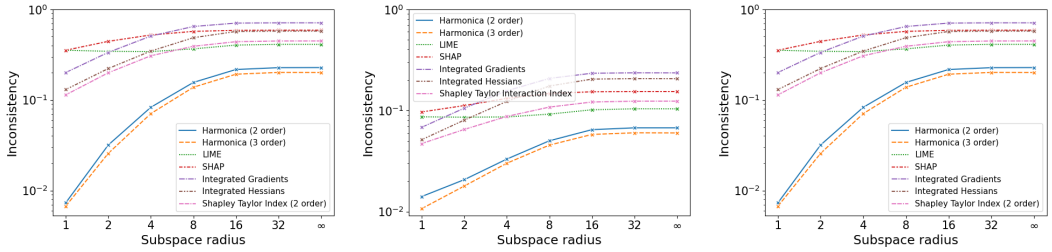
radius	L^2 norm	L^1 norm	L^0 norm
1	0.0149	0.0514	0.1301
2	0.0279	0.0804	0.2222
4	0.0507	0.1231	0.3466
8	0.0834	0.1746	0.4870
16	0.1008	0.2044	0.5704
32	0.1020	0.2067	0.5758
∞	0.1025	0.2064	0.5760

Integrated Hessians

radius	L^2 norm	L^1 norm	L^0 norm
1	0.0052	0.0471	0.1137
2	0.0090	0.0650	0.1995
4	0.0155	0.0874	0.3076
8	0.0232	0.1081	0.3915
16	0.0286	0.1214	0.4400
32	0.0295	0.1237	0.4478
∞	0.0295	0.1238	0.4482

Shapley Taylor Index^2

Table 2: The inconsistency of Harmonica and other baseline algorithms evaluated on the SST-2 dataset for different neighborhood with radius ranging from 1 to ∞ under L^2, L^1 and L^0 norm.



(a) L^2 norm (b) L^1 norm (c) L^0 norm

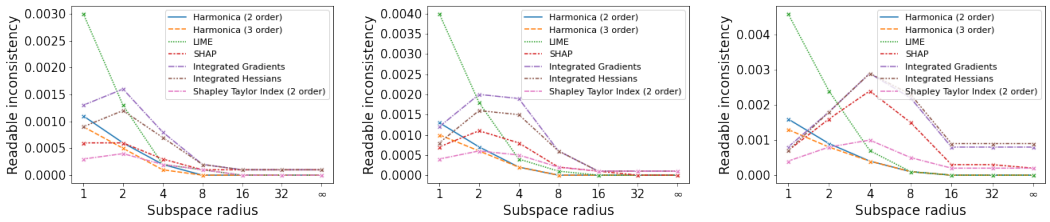
Figure 2: Visualization of inconsistency $\mathbb{I}_{\mathcal{D}}(f,g)$ evaluated on SST-2 dataset

Estimating Readable Inconsistency For convenience, we define the set of bases C^d up to degree d as $C^d = \{\chi_S | S \subseteq [n], |S| \leq d\}$. We evaluate the readable inconsistency on the set of bases C^3, C^2, and C^1. By definition in Eqn.(2), we have

$$\mathbb{R}_C(f, g) = \sum_{\chi_S \in C} \langle f - g, \chi_S \rangle^2 = \left(\mathop{\mathbb{E}}_{x \sim \{-1,1\}^n} \left[(f(x) - g(x)) \sum_{\chi_S \in C} \chi_S(x) \right] \right)^2 .$$

Then we could perform a sampling-based estimation of readable inconsistency just like we did in estimating inconsistency. Worth mentioning that since the size of set C^d satisfies $|C^d| = \sum_{i=0}^{d} \binom{n}{i}$ and the max length of sentences $n^* = 50$, $\sum_{\chi_S \in C} \chi_S(x)$, as the summation function of orthonormal basis, is easy to compute on every sample $x \in \{-1, 1\}^n$ (for n^* very large, we could perform another sampling step on this summation function).

Results on Readable Inconsistency Figure 3 shows the readable inconsistency evaluated on SST-2 dataset for different neighborhood with radius ranging from 1 to ∞ with support set C^3, C^2 and C^1.



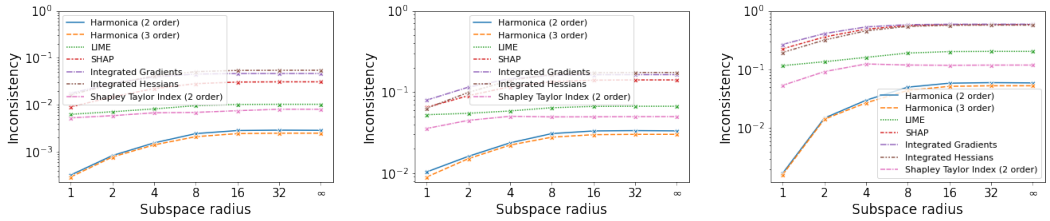
(a) bases C^3 (b) bases C^2 (c) bases C^1

Figure 3: Visualization of readable inconsistency $\mathbb{R}_C(f, g)$ evaluated on SST-2 dataset

6.2 Movie Review

The IMDb dataset contains long paragraphs, and each has many sentences. For the readability of results, we treat sentences as units instead of words – masking several words in a sentence may make the whole paragraph hard to understand and even meaningless, while masking a critical sentence has meaningful semantic effects. Therefore, the radius is defined as the maximum number of masked sentences. By default, we use periods, colons, and exclamations to separate sentences. The target network to be interpreted is a convolutional neural network (Kim, 2014) trained over this dataset.

Results on Inconsistency Figure 4 shows the inconsistency evaluated on IMDb dataset under the same settings of the SST-2 dataset, i.e., sampling-based estimation, with a slight modification of this procedure that on the IMDb dataset, masking operation is performed on sentences in one input paragraph instead (We also change the definition of radii accordingly). We can see that Harmonica consistently outperforms all the other baselines on all radii.



(a) L^2 norm (b) L^1 norm (c) L^0 norm

Figure 4: Visualization of inconsistency $\mathbb{I}_{\mathcal{D}}(f, g)$ evaluated on IMDb dataset

REFERENCES

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

Zeyuan Allen-Zhu and Yang Yuan. Improved svrg for non-strongly-convex or sum-of-non-convex objectives. In *International conference on machine learning*, pp. 1080–1089. PMLR, 2016.

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

Randall Balestriero. Neural decision trees. *arXiv preprint arXiv:1702.07360*, 2017.

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.

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

Ian Covert, Scott M Lundberg, and Su-In Lee. Understanding global feature contributions with additive importance measures. *Advances in Neural Information Processing Systems*, 33:17212–17223, 2020.

Ian Covert, Scott M Lundberg, and Su-In Lee. Explaining by removing: A unified framework for model explanation. *J. Mach. Learn. Res.*, 22:209–1, 2021.

Derek Doran, Sarah Schulz, and Tarek R Besold. What does explainable ai really mean? a new conceptualization of perspectives. *arXiv preprint arXiv:1710.00794*, 2017.

Edith Elkind, Leslie Ann Goldberg, Paul W Goldberg, and Michael Wooldridge. On the computational complexity of weighted voting games. *Annals of Mathematics and Artificial Intelligence*, 56(2): 109–131, 2009.

Eric Friedman and Herve Moulin. Three methods to share joint costs or surplus. *Journal of economic Theory*, 87(2):275–312, 1999.

Christopher Frye, Colin Rowat, and Ilya Feige. Asymmetric shapley values: incorporating causal knowledge into model-agnostic explainability. *Advances in Neural Information Processing Systems*, 33:1229–1239, 2020.

Michel Grabisch and Marc Roubens. An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of game theory*, 28(4):547–565, 1999.

Mark Hamilton, Scott Lundberg, Lei Zhang, Stephanie Fu, and William T Freeman. Axiomatic explanations for visual search, retrieval, and similarity learning. *arXiv preprint arXiv:2103.00370*, 2021.

Elad Hazan, Adam Klivans, and Yang Yuan. Hyperparameter optimization: A spectral approach. *arXiv preprint arXiv:1706.00764*, 2017.

Joseph D Janizek, Pascal Sturmfels, and Su-In Lee. Explaining explanations: Axiomatic feature interactions for deep networks. *Journal of Machine Learning Research*, 22(104):1–54, 2021.

Neil Jethani, Mukund Sudarshan, Ian Connick Covert, Su-In Lee, and Rajesh Ranganath. Fastshap: Real-time shapley value estimation. In *International Conference on Learning Representations*, 2021.

Yoon Kim. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882, 2014. URL `http://arxiv.org/abs/1408.5882`.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the fourier spectrum. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pp. 455–464, 1991.

Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. Faithful and customizable explanations of black box models. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 131–138, 2019.

Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, fourier transform, and learnability. *Journal of the ACM (JACM)*, 40(3):607–620, 1993.

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

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P11-1015`.

Yishay Mansour. Learning boolean functions via the fourier transform. In *Theoretical advances in neural computation and learning*, pp. 391–424. Springer, 1994.

Aria Masoomi, Davin Hill, Zhonghui Xu, Craig P. Hersh, Edwin K. Silverman, Peter J. Castaldi, Stratis Ioannidis, and Jennifer G. Dy. Explanations of black-box models based on directional feature interactions. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern recognition*, 65:211–222, 2017.

Ryan O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

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

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*, pp. 1135–1144, 2016.

Patrick Schwab and Walter Karlen. Cxplain: Causal explanations for model interpretation under uncertainty. *Advances in Neural Information Processing Systems*, 32, 2019.

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

LS Shapley. Quota solutions op n-person games1. *Edited by Emil Artin and Marston Morse*, pp. 343, 1953.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pp. 3145–3153. PMLR, 2017.

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

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

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1170.

Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665, 2014.

Guolong Su, Dennis Wei, Kush R Varshney, and Dmitry M Malioutov. Interpretable two-level boolean rule learning for classification. *arXiv preprint arXiv:1511.07361*, 2015.

Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. In *International conference on machine learning*, pp. 9269–9278. PMLR, 2020.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Mukund Sundararajan, Kedar Dhamdhere, and Ashish Agarwal. The shapley taylor interaction index. In *International conference on machine learning*, pp. 9259–9268. PMLR, 2020.

Michael Tsang, Sirisha Rambhatla, and Yan Liu. How does this interaction affect me? interpretable attribution for feature interactions. *Advances in neural information processing systems*, 33:6147–6159, 2020.

Berk Ustun and Cynthia Rudin. Methods and models for interpretable linear classification. *arXiv preprint arXiv:1405.4047*, 2014.

Berk Ustun, Stefano Traca, and Cynthia Rudin. Supersparse linear integer models for interpretable classification. *arXiv preprint arXiv:1306.6677*, 2013.

Guy Van den Broeck, Anton Lykov, Maximilian Schleich, and Dan Suciu. On the tractability of shap explanations. In *Proceedings of the 35th Conference on Artificial Intelligence (AAAI)*, 2021.

Jialei Wang, Ryohei Fujimaki, and Yosuke Motohashi. Trading interpretability for accuracy: Oblique treed sparse additive models. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1245–1254, 2015a.

Jiaxuan Wang, Jenna Wiens, and Scott M. Lundberg. Shapley flow: A graph-based approach to interpreting model predictions. In Arindam Banerjee and Kenji Fukumizu (eds.), *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pp. 721–729. PMLR, 2021a.

Rui Wang, Xiaoqian Wang, and David I Inouye. Shapley explanation networks. *arXiv preprint arXiv:2104.02297*, 2021b.

Tong Wang, Cynthia Rudin, Finale Doshi-Velez, Yimin Liu, Erica Klampfl, and Perry MacNeille. Or's of and's for interpretable classification, with application to context-aware recommender systems. *arXiv preprint arXiv:1504.07614*, 2015b.

Tong Wang, Cynthia Rudin, Finale Doshi-Velez, Yimin Liu, Erica Klampfl, and Perry MacNeille. A bayesian framework for learning rule sets for interpretable classification. *The Journal of Machine Learning Research*, 18(1):2357–2393, 2017.

Robert J Weber. Probabilistic values for games, the shapley value. essays in honor of lloyd s. shapley (ae roth, ed.), 1988.

Yongxin Yang, Irene Garcia Morillo, and Timothy M Hospedales. Deep neural decision trees. *arXiv preprint arXiv:1806.06988*, 2018.

Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. On explainability of graph neural networks via subgraph explorations. In *International Conference on Machine Learning*, pp. 12241–12252. PMLR, 2021.

Hao Zhang, Yichen Xie, Longjie Zheng, Die Zhang, and Quanshi Zhang. Interpreting multivariate shapley interactions in dnns. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 10877–10886. AAAI Press, 2021.

A PROOF FOR HARMONICA

Our proof is similar to the one in the original paper Hazan et al. (2017), with changes in the readable notion, which is now more flexible than being low order. First recall the classical Chebyshev inequality.

Theorem A.1 (Multidimensional Chebyshev inequality)**.** *Let X be an m dimensional random vector, with expected value $\mu = \mathbb{E}[X]$, and covariance matrix $V = \mathbb{E}\left[(X-\mu)(X-\mu)^T\right]$. If V is a positive definite matrix, for any real number $\delta > 0$:*

$$\mathbb{P}\left(\sqrt{(X-\mu)^T V^{-1}(X-\mu)} > \delta\right) \leq \frac{m}{\delta^2}$$

Proof of Theorem 4.1. Let f be an $(\varepsilon/4, s, C)$-bounded function written in the orthonormal basis as $\sum_S \hat{f}(S)\chi_S$. We can equivalently write f as $f = h + g$, where h is supported on C that only only includes coefficients of magnitude at least $\varepsilon/4s$ and the constant term of the polynomial expansion of f.

Since $L_1(f) = \sum_S \left|\hat{f}_S\right| \leq s$, we know h is $4s^2/\varepsilon + 1$ sparse. The function g is thus the sum of the remaining $\hat{f}(S)\chi_S$ terms not included in h. Denote the set of bases that appear in C but not in g as R, so we know the coefficient of f on the bases in R is at most $\epsilon/4s$.

Draw m (to be chosen later) random labeled examples $\left\{\left(z^1, y^1\right), \ldots, \left(z^m, y^m\right)\right\}$ and enumerate all $N = |C|$ basis functions $\chi_S \in C$ as $\{\chi_1, \ldots, \chi_N\}$. Form matrix A such that $A_{ij} = \chi_j\left(z^i\right)$ and consider the problem of recovering $4s^2/\varepsilon + 1$ sparse x given $Ax + e = y$ where x is the vector of coefficients of h, the i th entry of y equals y^i, and $e_i = g\left(z^i\right)$.

We will prove that with constant probability over the choice m random examples, $\|e\|_2 \leq \sqrt{\varepsilon m}$. Applying Theorem 5 in Hazan et al. (2017) by setting $\eta = \sqrt{\varepsilon}$ and observing that $\sigma_{4s^2/\varepsilon+1}(x)_1 = 0$ (see definition in the theorem), we will recover x' such that $\|x - x'\|_2^2 \leq c_2^2\varepsilon$ for some constant c_2. As such, for the function $\tilde{f} = \sum_{i=1}^N x_i'\chi_i$ we will have $\mathbb{E}\left[\|h - \tilde{f}\|^2\right] \leq c_2^2\varepsilon$ by Parseval's identity. Note, however, that we may rescale ε by constant factor $1/\left(2c_2^2\right)$ to obtain error $\varepsilon/2$ and only incur an additional constant (multiplicative) factor in the sample complexity bound. By the definition of g, we have

$$\|g\|^2 = \left(\sum_{S, \chi_S \notin C} \hat{f}(S)^2 + \sum_{S \in R} \hat{f}(S)^2\right) \tag{4}$$

where each $\hat{f}(S)$ for $S \in R$ is of magnitude at most $\varepsilon/4s$. By Fact 4 in Hazan et al. (2017) and Parseval's identity we have $\sum_R \hat{f}(R)^2 \leq \varepsilon/4$. Since f is $(\varepsilon/4, s, C)$-concentrated we have $\sum_{S, \chi_S \notin C} \hat{f}(S)^2 \leq \varepsilon/4$. Thus, $\|g\|^2$ is at most $\varepsilon/2$. Therefore, by triangle inequality $\mathbb{E}\left[\|f - \tilde{f}\|^2\right] \leq \mathbb{E}\left[\|h - \tilde{f}\|^2\right] + \mathbb{E}\left[\|g\|^2\right] \leq \varepsilon$. It remains to bound $\|e\|_2$. Note that since the examples are chosen independently, the entries $e_i = g\left(z^i\right)$ are independent random variables. Since g is a linear combination of orthonormal monomials (not including the constant term), we have $\mathbb{E}_{z \sim D}[g(z)] = 0$. Here we can apply linearity of variance (the covariance of χ_i and χ_j is zero for all $i \neq j$) and calculate the variance

$$\operatorname{Var}\left(g\left(z^i\right)\right) = \left(\sum_{S, \chi_S \notin C} \hat{f}(S)^2 + \sum_{S \in R} \hat{f}(S)^2\right)$$

With the same calculation as (4), we know $\mathrm{Var}\left(g\left(z^i\right)\right)$ *is at most* $\varepsilon/2$*. Now consider the covariance matrix* V *of the vector* e *which equals* $\mathbb{E}\left[ee^\top\right]$ *(recall every entry of* e *has mean 0). Then* V *is a diagonal matrix (covariance between two independent samples is zero), and every diagonal entry is at most* $\varepsilon/2$*. Applying Theorem A.1 we have*

$$\mathbb{P}\left(\|e\|_2 > \sqrt{\frac{\varepsilon}{2}}\delta\right) \leq \frac{m}{\delta^2}.$$

Setting $\delta = \sqrt{2m}$*, we conclude that* $\mathbb{P}\left(\|e\|_2 > \sqrt{\varepsilon m}\right) \leq \frac{1}{2}$*. Hence with probability at least* $1/2$*, we have that* $\|e\|_2 \leq \sqrt{\varepsilon m}$*. From Theorem 5 in Hazan et al. (2017), we may choose* $m = \tilde{O}\left(s^2/\varepsilon \cdot \log n^d\right)$*. This completes the proof. Note that the probability* $1/2$ *above can be boosted to any constant probability with a constant factor loss in sample complexity.*

For the running time complexity, we refer to Allen-Zhu & Yuan (2016) for optimizing linear regression with ℓ_1 *regularization. The running time is* $O((T\log\frac{1}{\epsilon} + L/\epsilon)\cdot|C|)$*, where* L *is the smoothness of each summand in the objective. Since each* χ_S *takes value in* $\{-1,1\}$*, the smoothness is bounded by the number of entries in each summand, which is* $|C|$*. Therefore, the running time is bounded by* $O((T\log\frac{1}{\epsilon} + |C|/\epsilon)\cdot|C|)$*.*

B Proof for Low-degree

Since we are given T samples to estimate $\hat{f}(S)$ for every S, we can directly apply the Hoeffding bound (notice that the function is bounded by B):

$$\Pr\left(|\alpha_S - \hat{f}(S)| \geq \epsilon\right)\cdot|C| \leq 2\exp\left(-\frac{2T\epsilon^2}{4B^2}\right) = 2\exp\left(-\frac{T\epsilon^2}{2B^2}\right)$$

Notice that $T \geq \frac{2B^2}{\epsilon^2}\log\frac{2|C|}{\delta}$, we know the right hand size is bounded by $\frac{\delta}{|C|}$, so Theorem 4.2 is proved.

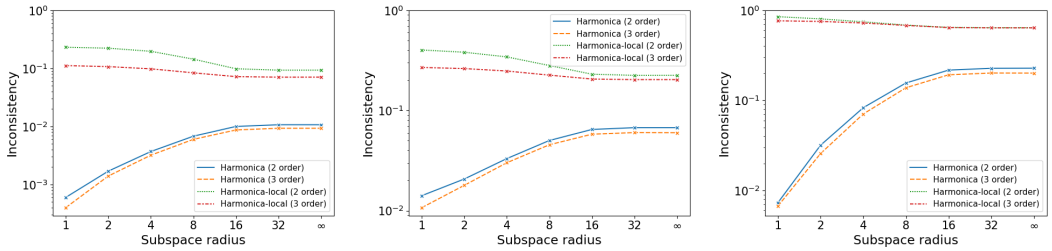
C More Experimental Results

C.1 Discussion on Harmonica-local

Notice that Harmonica recovers the function in the whole function space, but for our setting, if $\mathcal{N}_x$ is small, it suffices to recover a small neighborhood of the function. This leads to the idea that maybe we can apply Harmonica to a local space instead of the whole space. By doing that, we will have more concentrated samples in the local neighborhood. Therefore, minimizing the inconsistency in this neighborhood will be much easier. We call Harmonica with samples in the local neighborhood as Harmonica-local.

Interestingly, as shown in Figure 5, Harmonica-local does not work as expected. In fact, it works much worse than Harmonica and other methods. This is because Harmonica-local is based on a different sampling distribution, i.e., uniform distribution on $\mathcal{N}_x$. Theoretically, this space is not complete and does not have a well-defined inner product, so the Fourier basis cannot be directly applied, and all the previous theoretical guarantees do not hold. In summary, this experiment shows that following the theoretical assumptions is crucial for Harmonica.

C.2 Detailed Experimental Results



(a) L^2 norm (b) L^1 norm (c) L^0 norm

Figure 5: Visualization of inconsistency $\mathbb{I}_{\mathcal{D}}(f, g)$ evaluated on SST-2 dataset of algorithm Harmonica and Harmonica-local

radius	C_1	C_2	C_3
1	0.0016	0.0013	0.0011
2	0.0009	0.0007	0.0006
4	0.0004	0.0002	0.0002
8	0.0001	0.0000	0.0000
16	0.0000	0.0000	0.0000
32	0.0000	0.0000	0.0000
∞	0.0000	0.0000	0.0000

Harmonica^2

radius	C_1	C_2	C_3
1	0.0013	0.0010	0.0009
2	0.0008	0.0006	0.0005
4	0.0004	0.0002	0.0001
8	0.0001	0.0000	0.0000
16	0.0000	0.0000	0.0000
32	0.0000	0.0000	0.0000
∞	0.0000	0.0000	0.0000

Harmonica^3

radius	C_1	C_2	C_3
1	0.1459	0.1194	0.0935
2	0.0934	0.0645	0.0429
4	0.0366	0.0196	0.0088
8	0.0057	0.0018	0.0006
16	0.0005	0.0001	0.0001
32	0.0004	0.0001	0.0001
∞	0.0004	0.0001	0.0001

Harmonica-local^2

radius	C_1	C_2	C_3
1	0.0781	0.0625	0.0489
2	0.0513	0.0335	0.0212
4	0.0239	0.0098	0.0040
8	0.0050	0.0007	0.0002
16	0.0005	0.0001	0.0001
32	0.0004	0.0001	0.0001
∞	0.0003	0.0001	0.0001

Harmonica-local^3

radius	C_1	C_2	C_3
1	0.0046	0.0040	0.0030
2	0.0024	0.0018	0.0013
4	0.0007	0.0004	0.0002
8	0.0001	0.0001	0.0001
16	0.0000	0.0000	0.0000
32	0.0000	0.0000	0.0000
∞	0.0000	0.0000	0.0000

LIME

radius	C_1	C_2	C_3
1	0.0007	0.0007	0.0006
2	0.0016	0.0011	0.0006
4	0.0024	0.0008	0.0003
8	0.0015	0.0002	0.0001
16	0.0003	0.0001	0.0001
32	0.0003	0.0000	0.0001
∞	0.0002	0.0000	0.0001

SHAP

radius	C_1	C_2	C_3
1	0.0008	0.0012	0.0013
2	0.0018	0.0020	0.0016
4	0.0029	0.0019	0.0008
8	0.0022	0.0006	0.0002
16	0.0008	0.0001	0.0001
32	0.0008	0.0001	0.0001
∞	0.0008	0.0001	0.0001

Integrated Gradients

radius	C_1	C_2	C_3
1	0.0007	0.0008	0.0009
2	0.0018	0.0016	0.0012
4	0.0029	0.0015	0.0007
8	0.0023	0.0006	0.0002
16	0.0009	0.0001	0.0001
32	0.0009	0.0001	0.0001
∞	0.0009	0.0001	0.0001

Integrated Hessians

radius	C_1	C_2	C_3
1	0.0004	0.0004	0.0003
2	0.0008	0.0006	0.0004
4	0.0010	0.0005	0.0002
8	0.0005	0.0002	0.0001
16	0.0002	0.0001	0.0000
32	0.0002	0.0001	0.0000
∞	0.0002	0.0001	0.0000

Shapley Taylor Index^2

Table 3: The readable inconsistency $\mathbb{R}_C(f, g)$ of Harmonica and other baseline algorithms evaluated on the SST-2 dataset for different neighborhood with radius ranging from 1 to ∞ with support set C^3, C^2 and C^1.