

---

# Regularizing Black-box Models for Improved Interpretability

---

Gregory Plumb  
CMU

Maruan Al-Shedivat  
CMU

Eric Xing  
CMU

Ameet Talwalkar  
CMU

## Abstract

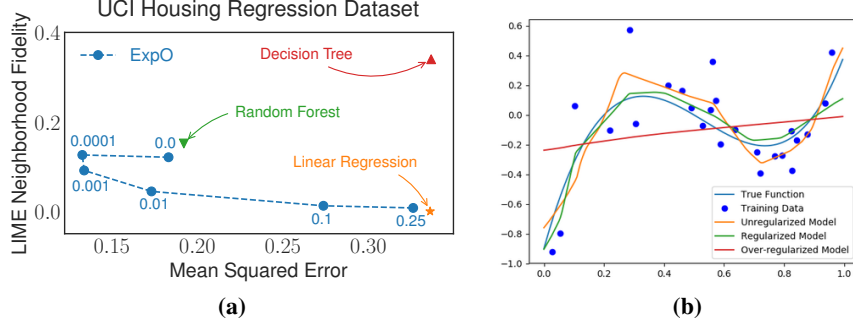
Most of the work on interpretable machine learning has focused on designing either inherently interpretable models, which typically trade-off accuracy for interpretability, or post-hoc explanation systems, which lack guarantees about their explanation quality. We propose an alternative to these approaches by directly regularizing a black-box model for interpretability at training time. Our approach explicitly connects three key aspects of interpretable machine learning: (i) the model’s innate explainability, (ii) the explanation system used at test time, and (iii) the metrics that measure explanation quality. Our regularization results in substantial improvement in terms of the explanation fidelity and stability metrics across a range of datasets and black-box explanation systems while slightly improving accuracy. Further, if the resulting model is still not sufficiently interpretable, the weight of the regularization term can be adjusted to achieve the desired trade-off between accuracy and interpretability. Finally, we justify theoretically that the benefits of explanation-based regularization generalize to unseen points.

## 1 Introduction

Complex learning-based systems are increasingly shaping our daily lives, and, in order to monitor and understand these systems, we require clear explanations of model behavior. While model interpretability has many definitions and is often largely application specific [1], local explanations are a popular and powerful tool [2]. Recent work on local interpretability in machine learning ranges from proposals of new models that are interpretable *by-design* [e.g., 3, 4] to model-agnostic, *post-hoc* algorithms for interpreting complex, black-box predictors such as ensembles and deep neural networks [e.g., 2, 5–8]. Despite the variety of technical approaches, the underlying goal of all of these works is to develop an interpretable predictive system that produces two outputs: a prediction and its underlying explanation.

Both interpretability by-design and post-hoc explanation strategies have limitations. On the one hand, the by-design approaches are restricted to working with model families that provide inherent explainability, potentially at the cost of accuracy. On the other hand, by performing two disjointed steps, there is no guarantee that post-hoc explainers applied to an arbitrary model will produce explanations of suitable quality. Moreover, recent approaches that claim to overcome this apparent trade-off between prediction accuracy and explanation quality are in fact by-design proposals that impose certain constraints on the underlying model families they consider [e.g., 9–11]. In this work, we propose a novel alternative strategy called *Explanation-based Optimization* (ExpO) that aims to address both of these shortcomings by adding an *interpretability regularizer* to the loss function of an arbitrary predictive model. A small demo of how our regularizer can influence the explainability and accuracy of a model is in Figure 1a.

**Illustration.** To motivate ExpO, consider a situation where Bob’s loan application is denied by a machine learning system (see Figure 1b for a toy illustration). In this setting, a good local explanation can help Bob understand how to improve his application in order to get the loan. Unfortunately, as



**Figure 1:** (a) Neighborhood Fidelity of LIME-generated explanations (lower is better) vs. predictive error of several models trained on the UCI Housing regression dataset. The values in blue denote regularization weight. (b) The effects of ExpO on a model predicting hypothetical credit rating. The abrupt kinks in the unregularized model make local linear approximations both less faithful to the model and less stable to small perturbations. The regularized model is much smoother and therefore easier to explain.

we see from Figure 1b, a standard model—a multi-layer perceptron trained with SGD—is difficult to explain well because it has many kinks and abrupt changes. Indeed, we can quantitatively measure the quality of local explanations using the standard *fidelity* [2, 10] and *stability* [11] explanation metrics. To make the learned model more amenable to local explanation, ExpO augments the objective function with fidelity- or stability-based regularizers, effectively controlling the degree of local explainability.

The specific contributions of our work are as follows:

1. **Interpretability Regularizers.** We introduce two explanation regularizers associated with the fidelity and stability explanation metrics. The first, ExpO-Fidelity, is designed for semantic features and explainers that directly make predictions, such as [2, 6, 10]. The second, ExpO-Stability, is tailored for non-semantic features (e.g., pixels) and explainers such as saliency maps [12], which identify features that are influential on a prediction. Both regularizers are differentiable and can be used to augment the objective function of an arbitrary model. In Section 3.1, we discuss how they differ from the classical approaches for local approximation and function smoothing.
2. **Generalizable Explanation Quality.** We analyze the properties of the explanation quality metrics and show that the benefits of our regularization generalize to unseen points. Specifically, we derive a bound on the gap between the fidelity of explanations on training and held out points and connect it with the local variance of the learned model.

**Empirical Results.** We evaluate models trained with and without the proposed regularizers on a variety of regression and classification tasks with semantic and image features.<sup>1</sup> We show experimentally that our regularizers slightly improve predictive performance across the nine datasets we consider (seven UCI regression tasks, a medical classification task, and MNIST). Moreover, from an interpretability perspective, our results demonstrate significant improvement in terms of explanation quality as measured by the fidelity and stability metrics. In particular, our regularization technique improved explanation fidelity by at least 25% on the UCI datasets and on the medical classification task; stability on MNIST was improved by orders of magnitude.

## 2 Background and Related Work

In this section, we introduce our notation, provide the necessary background on local explanations, and review the previous work that is most closely related to ExpO.

Consider a supervised learning problem, where our goal is to estimate a model,  $f : \mathcal{X} \mapsto \mathcal{Y}$ ,  $f \in \mathcal{F}$ , that maps input feature vectors,  $x \in \mathcal{X}$ , to targets,  $y \in \mathcal{Y}$ , and is trained using data,  $\{x_i, y_i\}_{i=1}^N$ . If the class of functions used for modeling the data is complex, we can understand the behavior of  $f$  in

<sup>1</sup>The code for our regularizers and all experiments is at: <https://anonymous.4open.science/r/73f2fe15-6041-4c7a-8083-69b10a76ad17/>

some neighborhood,  $N_x \in \mathcal{P}[\mathcal{X}]$  (where  $\mathcal{P}[\mathcal{X}]$  is the space of probability distributions over  $\mathcal{X}$ ), by generating a local *explanation*. We denote algorithms that produce explanations (*i.e.*, *explainers*) as  $e : \mathcal{X} \times \mathcal{F} \mapsto \mathcal{E}$ , where  $\mathcal{E}$  is the set of possible explanations. The choice of  $\mathcal{E}$  generally depends on whether or not  $\mathcal{X}$  consists of *semantic features*, and will be defined more precisely next.

## 2.1 Semantic Features

We call features *semantic* if people can reason about them and understand what it means when their values change (*e.g.*, a person’s income, the concentration of a chemical, etc.). Consequently, local explanations try to predict how the model’s output would change if the input was perturbed [2, 6, 10]. Thus, we can define the output space of the explainer as  $\mathcal{E}_s := \{g \in \mathcal{G} \mid g : \mathcal{X} \mapsto \mathcal{Y}\}$ , where  $\mathcal{G}$  is a class of interpretable (typically linear) functions.

**Fidelity-Metric.** When the explainer’s output space is  $\mathcal{E}_s$ , the explanation is defined as a function  $g : \mathcal{X} \mapsto \mathcal{Y}$ , and it is natural to evaluate how accurately  $g$  models  $f$  in a neighborhood  $N_x$  [2, 10]:

$$F(f, g, N_x) := \mathbb{E}_{x' \sim N_x} [(g(x') - f(x'))^2], \quad (1)$$

which we refer to as the *neighborhood-fidelity* (NF) metric. This metric is sometimes evaluated with  $N_x$  as a point mass on  $x$  and we call this version the *point-fidelity* (PF) metric. While Plumb et al. [10] argued that point-fidelity can be misleading because it does not measure generalization of  $e(x, f)$  across  $N_x$ , it has been used for evaluation in the prior work [2, 6, 13] and we report it in our experiments along with the neighborhood-fidelity for completeness.

**Black-box Explanation Systems.** Various explainers have been proposed to generate local explanations of the form  $g : \mathcal{X} \mapsto \mathcal{Y}$ , typically assuming that  $g$  is linear. In particular, LIME [2], one of the most popular black-box explanation systems<sup>2</sup>, solves the following optimization problem:

$$e(x, f) := \arg \min_{g \in \mathcal{E}_s} F(f, g, N_x) + \Omega(g), \quad (2)$$

where  $\Omega(e)$  stands for an additive regularizer that encourages certain desirable properties of the explanations (*e.g.*, sparsity). LIME’s objective function is closely related to the fidelity metric and subsequently to our proposed ExpO-Fidelity regularizer. Consequently, we expect our regularizer to improve the quality of LIME-generated explanations and our experimental results in Section 4.1 corroborate this hypothesis.

Along with LIME, we consider another black-box explanation tool, called MAPLE [10]. It differs substantially from LIME in that its *neighborhood function is learned from the data* rather than specified as a parameter. In our experiments, we evaluate the quality of MAPLE-generated local explanations for models regularized via ExpO-Fidelity, but do not use MAPLE’s learned neighborhood function to define ExpO-Fidelity. We view this as a good test case to see how optimizing the fidelity metric for one neighborhood generalizes to another one (see Section 3 for a more detailed discussion of this point). In Section 4.1, we see that regularizing for LIME neighborhoods improves MAPLE’s explanation quality.

## 2.2 Non-Semantic Features

Non-semantic features lack an inherent interpretation, with images being a canonical example.<sup>3</sup> When  $\mathcal{X}$  consists of non-semantic inputs, we cannot assign meaning to the difference between  $x$  and  $x'$ , hence it does not make sense to explain the difference between the predictions  $f(x)$  and  $f(x')$ . As a result, fidelity is not an appropriate explanation metric. Instead, in this context, local explanations try to identify which parts of the input are particularly influential on a prediction [15]. Consequently, we consider explanations of the form  $\mathcal{E}_{ns} := \mathbb{R}^d$ , where  $d$  is the number of features in  $\mathcal{X}$ .

<sup>2</sup>SHAP [6] is a popular variation of LIME that proposes a theoretically-motivated neighborhood sampling function, but requires explanations to be linear models that act on binary features. This requirement is too limiting in our case, hence SHAP is not used in our study.

<sup>3</sup>In general, it is not clear how to interpret perturbations on the pixel level or whether such perturbations result in ‘real’ images. However, in certain cases such as scientific imaging [14], each pixel value may have a precise meaning because of the way the images are processed.

---

**Algorithm 1** ExpO-Fidelity Regularizer

---

**input**  $f_\theta, x, N_x^{\text{reg}}, m$ 

- 1: Sample points:  $x'_1, \dots, x'_m \sim N_x^{\text{reg}}$
- 2: Compute predictions:

$$\hat{y}_j(\theta) = f_\theta(x'_j) \text{ for } j = 1, \dots, m$$

- 3: Produce a local linear explanation:

$$\beta_x(\theta) = \arg \min_{\beta} \sum_{j=1}^m (\hat{y}_j(\theta) - \beta^\top x'_j)^2$$

**output**  $\frac{1}{m} \sum_{j=1}^m (\hat{y}_j(\theta) - \beta_x(\theta)^\top x'_j)^2$ 

---

---

**Algorithm 2** ExpO-Stability Regularizer

---

**input**  $f_\theta, x, N_x^{\text{reg}}, m$ 

- 1: Sample points:  $x'_1, \dots, x'_m \sim N_x^{\text{reg}}$
- 2: Compute predictions:

$$\hat{y}_j(\theta) = f_\theta(x'_j), \text{ for } j = 1, \dots, m$$

**output**  $\frac{1}{m} \sum_{j=1}^m (\hat{y}_j(\theta) - f_\theta(x))^2$ 

---

**Stability Metric and Saliency Maps.** When the explainer’s output space is  $\mathcal{E}_{ns}$ , the explanation is a vector in  $\mathbb{R}^d$ , and cannot be directly compared to the underlying model itself, as in the case of the fidelity metric. Instead, the focus in this setting is on the degree to which the explanation changes between points in a local neighborhood, which we measure using the *stability metric* [11]:

$$\mathcal{S}(f, e, N_x) := \mathbb{E}_{x' \sim N_x} [\|e(x, f) - e(x', f)\|_2^2] \quad (3)$$

Various explainers [15–19] have been proposed to generate local explanations in  $\mathcal{E}_{ns}$ , with *saliency maps* [12] being a popular approach that we consider in this work. Saliency maps assign importance weights to image pixels based on the magnitude of the gradient of the predicted class with respect to the corresponding pixels.

Recent work on model interpretability emphasizes that more stable explanations tend to be more trustworthy [11, 20, 21]. Note that the stability metric can also be considered in the context of semantic features in addition to the fidelity metric, and we consider both in our experiments.

### 2.3 Related Methods

A few recently proposed approaches to model interpretability are closely related to our work. First, self-explaining neural networks (SENN) [11] (a variation of contextual explanation networks (CEN) [9]) is an interpretable by-design approach that additionally (indirectly) optimizes their models to produce stable explanations. Second, “Right For The Right Reasons” (RTFR) [22] selectively penalizes gradients of the output with respect to certain input features at some points to discourage their use by the model. Finally, a work concurrent with ours [23] proposed to regularize models of structured data to encourage explainability in a way that is similar to ExpO.

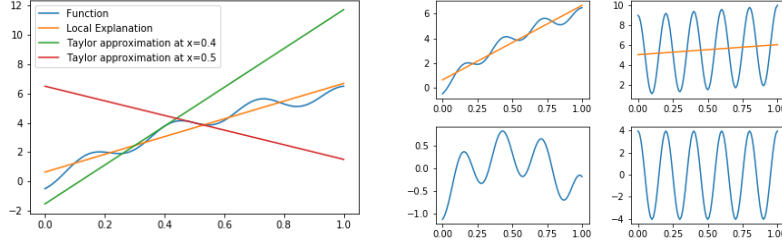
From a technical standpoint, SENN and RTFR both assume that the local explanation is close to the first order Taylor approximation of the model at that point. In Section 3.1, we demonstrate how Taylor approximations are often quite different from and more difficult to use than the neighborhood-based local explanations that we use in ExpO. Further, SENN’s regularizer requires the neural network to have a very particular structure and, therefore, unlike ExpO, cannot be applied to an arbitrary model. While RTFR’s regularization can be used with arbitrary models, it is not directly related to a measure of explanation quality; on the other hand, ExpO aims to directly improve quality of explanations with respect to a specific metric.

## 3 Explanation Optimization

Running black-box explainers on arbitrary models does not guarantee the quality of the produced explanations. To address this, we define a regularizer that can be added to the loss function and used to train an arbitrary model  $f$ . Specifically, we want to solve the following optimization problem:

$$\hat{f} := \arg \min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N (\mathcal{L}(f, x_i, y_i) + \gamma \mathcal{R}(f, N_{x_i}^{\text{reg}})) \quad (4)$$

where  $\mathcal{L}(f, x_i, y_i)$  is a standard predictive loss (e.g., squared error for regression or cross-entropy for classification),  $\mathcal{R}(f, N_{x_i}^{\text{reg}})$  is a regularizer that encourages explainability of  $f$  in the neighborhood of  $x_i$ , and  $\gamma > 0$  controls the regularization strength.



**Figure 2: Left:** A function (blue), its first order Taylor approximations at  $x = 0.4$  (green) and  $x = 0.5$  (red), and a local explanation of the function (orange) computed with  $x = 0.5$  and  $N_x = [0, 1]$ . **Right (top row):** Two functions (blue) and their local linear explanations (orange). The local explanations were computed with  $x = 0.5$  and  $N_x = [0, 1]$ . **Right (bottom row):** The unexplained portion of the function (residuals).

We define  $\mathcal{R}(f, N_x^{\text{reg}})$  based on either the neighborhood-fidelity, Eq. (1), or the neighborhood-stability, Eq. (3). In order to compute these metrics exactly, we would need to run an explainer algorithm,  $e$ ; this may be non-differentiable or too computationally expensive to use as a regularizer. Thus, for ExpO-Fidelity, we approximate  $e$  using a local linear model fit on points sampled from  $N_x^{\text{reg}}$  (Algorithm 1). For ExpO-Stability, we simply require that the model’s output not change too much across  $N_x^{\text{reg}}$  (Algorithm 2).<sup>4</sup>

To define a good regularization neighborhood,  $N_x^{\text{reg}}$ , requires taking the following into consideration. On the one hand, we would like  $N_x^{\text{reg}}$  to be similar to  $N_x$ , as used in Eq. 1 or Eq. 3, so that the neighborhoods used for regularization and for evaluation match. On the other hand, we also would like  $N_x^{\text{reg}}$  to be consistent with the ‘local neighborhood’ defined by  $e$  internally, which may differ from  $N_x$ . For LIME, this is not a problem since the internal definition of the ‘local neighborhood’ is a hyperparameter that we can set. However for MAPLE, the ‘local neighborhood’ is learned from the data, and hence the regularization and explanation neighborhoods may differ. Ultimately, we left resolving this tension to future work.

**Computational Cost.** Algorithm 1 could be prohibitively expensive since the number of samples,  $m$ , from  $N_x^{\text{reg}}$ , has to be proportional to the dimension of  $x$ , resulting in  $O(d^3)$  operations to compute the regularizer for a given point. In addition to running experiments with ExpO-Fidelity, we run experiments with a randomized version of the Algorithm 1 that randomly selects one dimension of  $x$  to perturb according to  $N_x^{\text{reg}}$  and penalizes the error of a local linear model along that dimension. This breaks the dependence of the computational cost of the objective function on the dimension of  $x$  (bringing it back to  $O(1)$ ) and allows us to compute each gradient step with some constant increase in the number of function evaluations. We call this variation ExpO-1D-Fidelity.

### 3.1 Understanding the Properties of ExpO

The goal of this section is to compare the behavior of local linear explanations and our regularizer to some existing theoretical function approximations and measures of variance to help develop an intuitive understanding of ExpO. First, we compare neighborhood-based local linear explanations to first order Taylor approximations to show that they can have fundamentally very different behaviors. Second, we compare ExpO-Fidelity to the Lipschitz Constant (LC) and Total Variation (TV) of the learned function.

**Local Explanation vs. Taylor Approximations.** A natural question to ask is, *Why should we sample from  $N_x$  in order to locally approximate  $f$  when there are easier and theoretically motivated approximations?* One possible way to do this is via the Taylor approximation [11]. The downside of a Taylor approximation-based approach is that such approximation cannot readily be adjusted to different neighborhood scales and its fidelity and stability strictly depend on the learned function. This can be seen in Figure 2 where the Taylor approximations at two nearby points are both radically different and not faithful to the model outside of an infinitesimal neighborhood.

**Fidelity-Regularization and the Model’s LC or TV.** From a theoretical perspective, our regularizer is similar to controlling the Lipschitz Constant or Total Variation of  $f$  across  $N_x$  after removing

<sup>4</sup>We note that a similar procedure was explored previously in [24] for adversarial robustness.

the part of  $f$  explained by  $e(x, f)$ . From an interpretability perspective, there is nothing inherently wrong with having a large LC or TV, which is demonstrated in Figure 2. However, once we take into account what can be explained by  $e(x, f)$ , then upper bounding any one of ExpO-Fidelity, the LC, or the TV will upper bound the remaining ones.

### 3.2 Generalization of Local Linear Explanations

To conclude our analysis, we study the quality of local linear explanations in terms of generalization. Note that ExpO regularization encourages learning models that are explainable in the neighborhoods of each *training point*. However, how would this property generalize to unseen points?

We answer this question by providing a generalization bound in terms of neighborhood-fidelity metric. First, we assume that local linear explanations,  $\beta_x$ , are obtained by solving the ordinary least squares regression problem (as given in Algorithm 1). The fidelity of the explanation in expectation over the neighborhood  $N_x$  can be computed analytically:

$$r(f, x) = \mathbb{E}_{N_x} [f(x')^2] - \mathbb{E}_{N_x} [f(x')x']^\top \mathbb{E}_{N_x} [[x'x'^\top]]^{-1} \mathbb{E}_{N_x} [f(x')x'] \leq \text{Var}_{N_x} [f(x')] \quad (5)$$

where expectation  $\mathbb{E}_{N_x} [\cdot]$  is taken with respect to  $x'$  over the neighborhood  $N_x$ . Note the equality in (5) is the expected value of the squared residual between  $f(x)$  and the optimal local linear explanation, which is upper-bounded by the variance of the model in the corresponding neighborhood.

For the explanations to generalize, we would like to make sure that the gap between the average fidelity on the training set and the expected fidelity is small with high probability. More formally, the following inequality should hold:

$$\mathbb{P} \left( \mathbb{E} [r(f, x)] - \frac{1}{n} \sum_{i=1}^n r(f, x_i) > \varepsilon \right) < \delta_n(\varepsilon) \quad (6)$$

Under certain mild assumptions on the local behavior of  $f(x)$ , the following proposition specifies a particular bound. Further, we show empirically that the benefits of our novel regularizers on explanation quality provably generalize to unseen test points.

**Proposition 1** *Let the neighborhood sampling function  $N_x$  be characterized by some parameter  $\sigma$  (e.g., the effective radius of a neighborhood) and the variance of the trained model  $f(x)$  across all such neighborhoods be bounded by some constant  $C(\sigma) > 0$ . Then, the following bound holds with at least  $1 - \delta$  probability:*

$$\mathbb{E} [r(f, x)] \leq \frac{1}{n} \sum_{i=1}^n r(f, x_i) + \sqrt{\frac{C^2(\sigma) \log \frac{1}{\delta}}{2n}} \quad (7)$$

**Proof.** (Sketch) By assumption, the variance of the model  $f(x)$  is bounded in each local neighborhood specified by  $N_x$ . Then (5) implies that each residual is bounded as  $0 \leq r(f, x) \leq C(\sigma)$ . The result then follows by applying Hoeffding’s inequality and rearranging the terms. ■

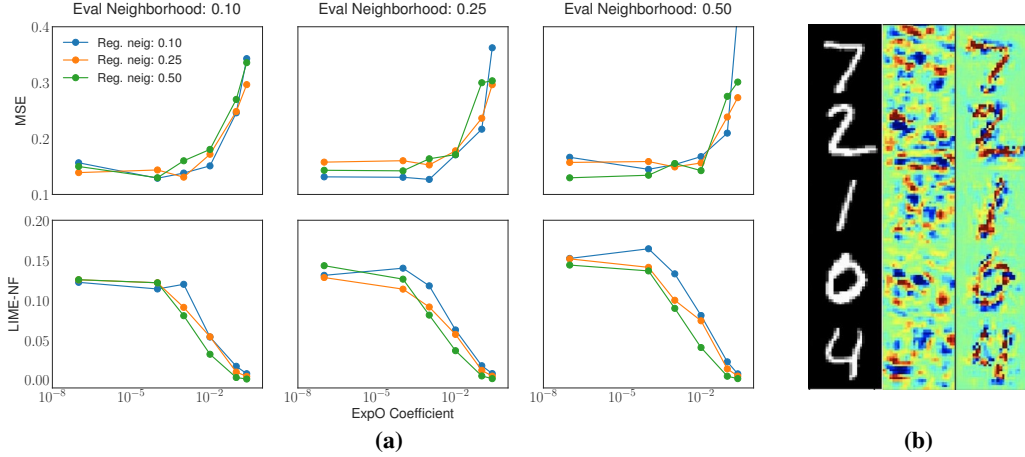
## 4 Experimental Results

In our first set of experiments, we demonstrate the effectiveness of ExpO-Fidelity and ExpO-1D-Fidelity on datasets with semantic features using several regression problems from the UCI collection [25] as well as an in-hospital mortality classification problem.<sup>5</sup> Our second experiment demonstrates the effectiveness of ExpO-Stability for creating saliency maps [12] on MNIST [26]. Dataset statistics are given in Table 1.

**Table 1:** Statistics of the datasets.

Dataset	# samples	# dims
autompgs	392	8
communities	1993	103
day	731	15
housing	506	12
music	1059	70
winequality-red	1599	12
YearPredictionMSD	515345	90
SUPPORT2	9104	51
MNIST	60000	784

<sup>5</sup><http://biostat.mc.vanderbilt.edu/wiki/Main/SupportDesc>.



**Figure 3: (a)** A comparison showing the effects of the  $\sigma$  parameter of  $N_x$  and  $N_x^{reg}$  on the UCI Housing dataset. The LIME-NF metric grows slowly with  $\sigma$  for  $N_x$  as expected. Despite being very large, using  $\sigma = 0.5$  for  $N_x^{reg}$  is generally best for the LIME-NF metric and possibly for accuracy. **(b)** Original MNIST images (left) and saliency maps of an unregularized (middle) and regularized (right) models.

#### 4.1 Neighborhood-Fidelity Regularization

First, we compare models trained without our regularizers to models trained with them. We report accuracy and three interpretability metrics: (1) Point-Fidelity (PF), (2) Neighborhood-Fidelity (NF), (3) Stability (S) for explanations generated by LIME and MAPLE. For example, the “MAPLE-PF” label corresponds to the Point-Fidelity Metric for explanations produced by MAPLE.

**Experimental Setup.** The network architectures and hyper-parameters were chosen by a simple grid search. All inputs were standardized to have mean zero and variance one (including the response variable for regression problems). For the final set of experiments, we set  $N_x$  to be  $\mathcal{N}(x, \sigma)$  with  $\sigma = 0.1$ . Analysis of the effects of different neighborhood sizes is given in Figure 3a and shows that the size is not critical (the value of LIME-NF increase only slightly with  $\sigma$ ). For the UCI regression datasets and the in-hospital mortality classification task, we set  $N_x^{reg}$  to be  $\mathcal{N}(x, \sigma)$  with  $\sigma = 0.5$  as we found this to produce slightly more accurate and more interpretable models (Figure 3a).

**UCI Regression Experiments.** The effects of ExpO-Fidelity and ExpO-1D-Fidelity on model accuracy and interpretability metrics are in Table 2. ExpO-Fidelity frequently improved the interpretability metrics by over 50%, with the smallest improvements being around 25%. In fact, our regularization lowered the prediction error on the ‘communities’, ‘day’, and ‘YearPredictionMSD’ datasets, which lets us conclude that it has a small positive effect on accuracy as well as a substantial benefit to the interpretability metrics. ExpO-1D-Fidelity, while generally having a similar effect, consistently improve interpretability of the models.

We also run experiments on the ‘YearPredictionMSD’ dataset<sup>6</sup> to understand the scalability of ExpO to larger tasks. However, MAPLE-based evaluation was fairly slow on this dataset, and hence we only evaluate the interpretability metrics with respect to LIME on the first 1000 testing points. Both ExpO-Fidelity and ExpO-1D-Fidelity improved LIME’s interpretability metrics at least 50% and both improved the model accuracy.

**Medical Classification Experiments.** The SUPPORT2 dataset is used for in-hospital mortality prediction. The output layer of our models is the softmax over logits for two classes. Consequently, we run each explanation system on each of the individual logits. Table 3 presents the results. Again, we observe that ExpO-Fidelity did not affect the accuracy but did improve the interpretability metrics by 50% or more. ExpO-1D-Fidelity slightly decreased accuracy and did not improve the interpretability metrics by as much as ExpO-Fidelity, but it did improve them by at least 25%.

<sup>6</sup>The task is to predict release year of song from a set of acoustic features, treated as a regression problem as in Bloniarz et al. [27]; the dataset is denoted MSD in Table 2.

**Table 2:** Uregularized model vs. the same model trained with ExpO-Fidelity or ExpO-1D-Fidelity on the UCI regression datasets. Results are shown across 20 trials (with the standard error in parenthesis). Statistically significant improvement ( $p = 0.05$ ) due to Fidelity is denoted in bold and due to 1D-Fidelity is underlined.

Metric	Regularizer	autompgs	communities	day <sup>†</sup> ( $10^{-3}$ )	housing	music	winequality.red	MSD
MSE	None	0.14 (0.03)	0.49 (0.05)	1.000 (0.300)	0.14 (0.05)	0.72 (0.09)	0.65 (0.06)	0.583 (0.018)
	Fidelity	0.13 (0.02)	<b>0.46 (0.03)</b>	<b>0.002 (0.002)</b>	0.15 (0.05)	0.67 (0.09)	0.64 (0.06)	<b>0.557 (0.0162)</b>
	1D-Fidelity	0.13 (0.02)	0.55 (0.04)	5.800 (8.800)	0.15 (0.07)	0.74 (0.07)	0.66 (0.06)	<u>0.548 (0.0154)</u>
LIME-PF	None	0.040 (0.011)	0.100 (0.013)	1.200 (0.370)	0.14 (0.036)	0.110 (0.037)	0.0330 (0.0130)	0.116 (0.0181)
	Fidelity	<b>0.011 (0.003)</b>	<b>0.080 (0.007)</b>	<b>0.041 (0.007)</b>	<b>0.057 (0.017)</b>	<b>0.066 (0.011)</b>	<b>0.0025 (0.0006)</b>	<b>0.0293 (0.00709)</b>
	1D-Fidelity	<u>0.029 (0.007)</u>	<u>0.079 (0.026)</u>	0.980 (0.380)	0.064 (0.017)	0.080 (0.039)	0.0029 (0.0011)	0.057 (0.0079)
LIME-NF	None	0.041 (0.012)	0.110 (0.012)	1.20 (0.36)	0.140 (0.037)	0.112 (0.037)	0.0330 (0.0140)	0.117 (0.0178)
	Fidelity	<b>0.011 (0.003)</b>	<b>0.079 (0.007)</b>	<b>0.04 (0.07)</b>	<b>0.057 (0.018)</b>	<b>0.066 (0.011)</b>	<b>0.0025 (0.0006)</b>	<b>0.029 (0.007)</b>
	1D-Fidelity	<u>0.029 (0.007)</u>	<u>0.080 (0.027)</u>	1.00 (0.39)	0.064 (0.017)	0.080 (0.039)	0.0029 (0.0011)	0.0575 (0.0079)
LIME-S	None	0.0011 (0.0006)	0.022 (0.003)	0.150 (0.021)	0.0047 (0.0012)	0.0110 (0.0046)	0.00130 (0.00057)	0.0368 (0.00759)
	Fidelity	<b>0.0001 (0.0003)</b>	<b>0.005 (0.001)</b>	<b>0.004 (0.004)</b>	<b>0.0012 (0.0002)</b>	<b>0.0023 (0.0004)</b>	<b>0.00007 (0.00002)</b>	<b>0.00171 (0.00034)</b>
	1D-Fidelity	<u>0.0008 (0.0003)</u>	<u>0.018 (0.008)</u>	<u>0.100 (0.047)</u>	<u>0.0025 (0.0007)</u>	0.0084 (0.0052)	0.00016 (0.00005)	0.0125 (0.00291)
MAPLE-PF	None	0.0160 (0.0088)	0.16 (0.02)	1.0000 (0.3000)	0.057 (0.024)	0.17 (0.06)	0.0130 (0.0078)	—
	Fidelity	<b>0.0014 (0.0006)</b>	<b>0.13 (0.01)</b>	<b>0.0002 (0.0003)</b>	<b>0.028 (0.013)</b>	0.14 (0.03)	<b>0.0027 (0.0010)</b>	—
	1D-Fidelity	<u>0.0076 (0.0038)</u>	<u>0.092 (0.03)</u>	<u>0.7600 (0.3000)</u>	<u>0.027 (0.012)</u>	0.13 (0.05)	0.0016 (0.0007)	—
MAPLE-NF	None	0.0180 (0.0097)	0.31 (0.04)	1.2000 (0.3200)	0.066 (0.024)	0.18 (0.07)	0.0130 (0.0079)	—
	Fidelity	<b>0.0015 (0.0006)</b>	<b>0.24 (0.05)</b>	<b>0.0003 (0.0004)</b>	<b>0.033 (0.014)</b>	<b>0.14 (0.03)</b>	<b>0.0028 (0.0010)</b>	—
	1D-Fidelity	<u>0.0084 (0.0040)</u>	<u>0.16 (0.05)</u>	<u>0.9400 (0.3600)</u>	<u>0.032 (0.013)</u>	0.14 (0.06)	0.0017 (0.0008)	—
MAPLE-S	None	0.0150 (0.0099)	1.2 (0.2)	0.0003 (0.0008)	0.18 (0.14)	0.08 (0.06)	0.0043 (0.0020)	—
	Fidelity	<b>0.0017 (0.0005)</b>	<b>0.8 (0.4)</b>	0.0004 (0.0004)	<b>0.10 (0.08)</b>	<b>0.05 (0.02)</b>	<b>0.0009 (0.0004)</b>	—
	1D-Fidelity	<u>0.0077 (0.0051)</u>	<u>0.6 (0.2)</u>	1.2000 (0.6600)	<u>0.09 (0.06)</u>	0.04 (0.02)	0.0004 (0.0002)	—

<sup>†</sup>The relationship between inputs and targets in UCI Day dataset is very close to linear and hence all errors are orders of magnitude smaller.

**Table 3:** Uregularized model vs. the same model trained with ExpO-Fidelity or ExpO-1D-Fidelity on the SUPPORT2 binary classification dataset. Each explanation metric computed for both positive and negative class logits. Results are shown across 10 trials (with the standard error in parenthesis). Improvement due to Fidelity and 1D-Fidelity over unregularized model is statistically significant ( $p = 0.05$ ) for each metric.

Output	Regularizer	LIME-PF	LIME-NF	LIME-S	MAPLE-PF	MAPLE-NF	MAPLE-S
Positive	None	0.177 (0.063)	0.182 (0.065)	0.0255 (0.0084)	0.024 (0.008)	0.035 (0.010)	0.34 (0.06)
	Fidelity	<b>0.050 (0.008)</b>	<b>0.051 (0.008)</b>	<b>0.0047 (0.0008)</b>	<b>0.013 (0.004)</b>	<b>0.018 (0.005)</b>	<b>0.13 (0.05)</b>
	1D-Fidelity	<u>0.082 (0.025)</u>	<u>0.085 (0.025)</u>	<u>0.0076 (0.0022)</u>	<u>0.019 (0.005)</u>	<u>0.025 (0.005)</u>	<u>0.16 (0.03)</u>
Negative	None	0.198 (0.078)	0.205 (0.080)	0.0289 (0.0121)	0.028 (0.010)	0.040 (0.014)	0.37 (0.18)
	Fidelity	<b>0.050 (0.008)</b>	<b>0.051 (0.008)</b>	<b>0.0047 (0.0008)</b>	<b>0.013 (0.004)</b>	<b>0.018 (0.005)</b>	<b>0.13 (0.03)</b>
	1D-Fidelity	<u>0.081 (0.026)</u>	<u>0.082 (0.027)</u>	<u>0.0073 (0.0021)</u>	<u>0.019 (0.006)</u>	<u>0.024 (0.007)</u>	<u>0.16 (0.06)</u>

Accuracy (%): None:  $83.0 \pm 0.3$ , Fidelity:  $83.4 \pm 0.4$ , 1D-Fidelity:  $82.0 \pm 0.3$ .

## 4.2 Stability Regularization

In this final experiment, we fit a convolutional neural network to MNIST and then evaluate the stability of its saliency maps to perturbations, where  $N_x \equiv N_x^{reg} := \text{Unif}(x - 0.05, x + 0.05)$ . Both an unregularized model and a model trained with ExpO-Stability achieved the accuracy of 99%. This demonstrates one of the practical differences between SENN [11] and our regularizers: SENN places strict structural constraints on the network and subsequently lowers the testing accuracy to roughly 97%; this is not the case for ExpO which can be applied to arbitrary networks. ExpO regularization decreased the average  $l_2$  distance between the explanation at  $x$  and some  $x' \sim N_x$  from 6.94 to 0.0008. Finally, our regularization makes the resulting saliency maps look much better qualitatively by focusing on the presence or absence of certain pen strokes as seen in Figure 3b.

## 5 Conclusion

In this work, we have introduced the novel idea of directly regularizing arbitrary models to be more interpretable. We contrasted our regularizers to classical approaches for function approximation and smoothing and provided a generalization bound for them. We demonstrated, across a variety of problem settings and explainers, that our regularizers slightly improve model accuracy and improve the interpretability metrics by somewhere from 25% to orders of magnitude.

We believe that potential future work may focus on three areas: (1) exploring alternative neighborhood functions,  $N_x^{reg}$ , that match those used by other black-box explanation systems, (2) exploring how to regularize for non-local interpretability metrics, and finally (3) exploring the interaction between the regularizers and the optimization process, e.g., progressively changing the importance of the regularization during training or using the regularization as an addition training step. The need for (3) is demonstrated in Figure 1a where we see that we cannot train a neural network to smoothly interpolate down to the accuracy and interpretability of a linear model.



## References

- [1] Zachary C Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.
- [2] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016.
- [3] Fulton Wang and Cynthia Rudin. Falling rule lists. In *Artificial Intelligence and Statistics*, pages 1013–1022, 2015.
- [4] Rich Caruana et al. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1721–1730. ACM, 2015.
- [5] Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155*, 2016.
- [6] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017.
- [7] 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*, pages 618–626, 2017.
- [8] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International Conference on Machine Learning*, pages 2673–2682, 2018.
- [9] Maruan Al-Shedivat, Avinava Dubey, and Eric P Xing. Contextual explanation networks. *arXiv preprint arXiv:1705.10301*, 2017.
- [10] Gregory Plumb, Denali Molitor, and Ameet S Talwalkar. Model agnostic supervised local explanations. In *Advances in Neural Information Processing Systems*, pages 2516–2525, 2018.
- [11] David Alvarez-Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. In *Advances in Neural Information Processing Systems*, pages 7785–7794, 2018.
- [12] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.
- [13] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. AAAI, 2018.
- [14] Luke de Oliveira, Michael Kagan, Lester Mackey, Benjamin Nachman, and Ariel Schwartzman. Jet-images—deep learning edition. *Journal of High Energy Physics*, 2016(7):69, 2016.
- [15] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org, 2017.
- [16] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.
- [17] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.
- [18] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

- [19] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, pages 9525–9536, 2018.
- [20] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. *arXiv preprint arXiv:1710.10547*, 2017.
- [21] David Alvarez-Melis and Tommi S Jaakkola. On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049*, 2018.
- [22] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717*, 2017.
- [23] Guang-He Lee, Wengong Jin, David Alvarez-Melis, and Tommi S Jaakkola. Functional transparency for structured data: a game-theoretic approach. *arXiv preprint arXiv:1902.09737*, 2019.
- [24] Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the robustness of deep neural networks via stability training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4480–4488, 2016.
- [25] Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017. URL <http://archive.ics.uci.edu/ml>.
- [26] Yann LeCun. The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>, 1998.
- [27] Adam Bloniarz, Ameet Talwalkar, Bin Yu, and Christopher Wu. Supervised neighborhoods for distributed nonparametric regression. In *Artificial Intelligence and Statistics*, pages 1450–1459, 2016.