Faithful Perturbations and Evaluations for Post-Hoc Local Explanation Methods

Iain N. Smith[†,*], Osmar R. Zaiane[†]
[†] University of Alberta, Edmonton, Canada

Abstract

There are many pre-trained AI models currently in use because of their high performance and quick responses on important tasks. With new laws and legislature passing that is aware and critical of these methods we now require explanations from these models. However, most of the original training data are no longer available and it's impractical to expend resources to train a new explainable model. An alternative is to produce explanations of these pre-trained models, known as post-hoc explanations. The most popular of these methods, LIME, has seen a great deal of employment on this problem, but it has unaddressed issues with performance. Something we presume is caused by the quality of the local data generated to train it, known as perturbations. To fix this issue and clarify the goals of post-hoc explanations we propose using a distribution that better follows patterns in the data when generating perturbed samples. A distribution is used to add noise or create more local samples, producing perturbations. Originally this is the normal distribution but there are many others. We evaluate each approach by estimating fidelity on real local-data using the nearest neighbors to the explained sample. We find that when there are correlations between features, using multivariate perturbations greatly improves generalizability, more so than other perturbation approaches. This works particularly well for complex post-hoc explainers, and when there is little to gain from interactions there is no noticeable decrease to performance to worry about.

Keywords: post-hoc explanations, perturbations, explainable artificial intelligence.

1. Introduction

Artificial intelligence (AI) models are continually being used in critical applications while governments and policy are only just starting to catch up. During the interim, models that explain their decision making, white-boxes, have been far out-performed, and out-numbered, by those that cannot be explained, black-boxes. Now, if most models are already trained then retraining them as explainable models is expensive or may not even be possible, if data is no longer available. An approach to address the limits of existing models we can instead explain the ones that are already trained, making post-hoc explanations. Some post-hoc explanation approaches rely on the idea that we can train a white-box to behave the same way as the black-box but on a simpler, local dataset based around a particular sample we want to explain. These methods are trained using artificial data produced by making small changes to the sample referred to as perturbations [1–3]. However, we are concerned that many of these perturbation approaches do not seem to align with the core values of explainable AI (XAI), since some rely on structures we generally use in black-boxes [2] or cannot be directly traced for reasoning over a perturbation's features [3]. Questions like these about the source of perturbations may reduce user trust in the explainer. Modern techniques tend to ignore this as the final explainer will perform well, but if the explainer ever performs poorly we should have the ability to blame or debug any cause of it, including the perturbations. In combination, one problem that has to be solved to further post-hoc explanations is producing trustworthy and confident evaluations with understandable perturbation approaches.

[*] ins@ualberta.ca

This article is © 2025 by author(s) as listed above. The article is licensed under a Creative Commons Attribution (CC BY 4.0) International license (https://creativecommons.org/licenses/by/4.0/legalcode), except where otherwise indicated with respect to particular material included in the article. The article should be attributed to the author(s) identified above.

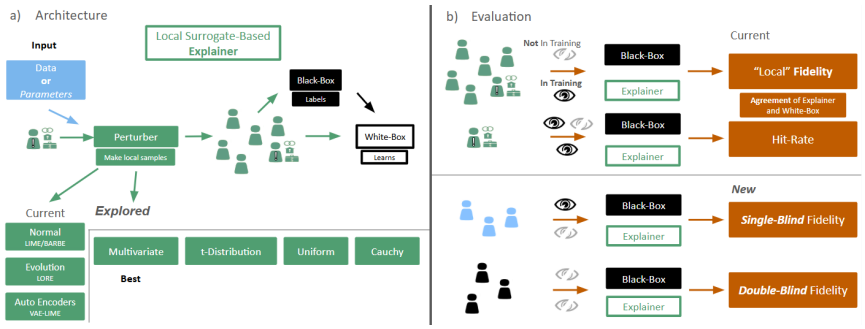


Figure 1. a) Architecture of a local surrogate-based explainer, we explore 4 new distributions that perturb data to produce training samples in comparison to existing approaches. b) We propose 2 new fidelity measures (bottom) using data unseen by the explainer, unlike "local" fidelity and hit-rate (top, which use data the explainer has seen). The new evaluation measures are weighted by distance from the sample to keep evaluation local.

Another goal central to XAI is having trustworthy white-box models. To understand trustworthiness we need to measure it. Typically this is done using fidelity, a measurement of how much the white-box explainer agrees with the black-box model. If the two models make the same correct predictions and incorrect predictions we expect them to be making decisions the same way. Some approaches may use fidelity with the perturbed data, data generated from the local sample. We expect this to be high since it may train on a large amount of perturbed data or even overfit to it. This can be a problem if the data does not represent behavior similar to the original dataset. Since unlikely samples will make the white-box fit to noise rather than general behaviour of the black box. The aim should then be evaluating on data in distribution, at least for experiments where we have it. Not only to have data used to train the black-box model, where we expect it to perform the best, but to include data unseen by both models to see how well the white-box generalizes.

We propose using an improved approach to making local perturbations. With the mechanism behind local sampling, and how the white-box uses it, remaining clear to the user. A proper perturbation method should also improve evaluations, not only how well a method can train on the data, but how well it generalizes to data local to the input sample that neither method has seen before. To serve this purpose we propose evaluating models using single-blind (the black-box was trained on the data) and double-blind (the black-box was not trained on the data, hold-out set) tests.

In Section 2 we discuss limitations of previous work which motivate our contributions:

- Producing a problem definition for local explanations that aligns with current approaches (in Section 3.3)
- Motivating two new evaluation criterion (weighted single-blind and double-blind tests in Sections 4.1 and 4.2)
- To improve results on these and other criteria we explore changes to sampling distributions for continuous and discrete features (in Sections 4.3 and 4.4)
- Concluding with robust experiments (described in Section 5) using these changes to re-evaluate the state-of-art and new local sampling (in Section 6)

2. Related Work

2.1. Post-Hoc Explanations

Some post-hoc explanations are done using models with human-interpretable features, called white-box models, trained on predictions made by the black-box model. The approach

was popularized by local interpretable model-agnostic explanations (LIME) [1], which explains a black-box based on each feature's contribution to a classification, known as feature importance. Feature importance comes from a ridge regression trained to closely match probability predictions of the black-box model. LIME perturbs input data using a univariate normal distribution with parameters determined by the training data distribution to produce perturbations. The data are produced over the whole distribution with locality determined by a kernel function used to weigh the samples during training.

There are some limits to LIME's perturbation strategy that methods like VAE-LIME [2] attempt to fix. Variational Auto-Encoders (VAEs) encode the training data into a latent space of values using a neural network architecture. The latent values can then be perturbed to generate new data by using a set of decoding layers similar to the original input structure.

Seemingly VAEs are an ideal solution, the data to be perturbed is in a lower dimensional subspace, and decoding should accurately represent the boundaries of the original data despite the starting conditions. One problem is that this always requires training data, while other approaches can have parameters fed in to produce perturbations. This is because a distribution's parameters can be interpreted by a user while neural network layers cannot. There may still be bias towards an explanation that functions for the highest density regions of the original sampling distribution. Even the choice of latent space size may cause issues with the actual quality of examples produced, as seen in examples of numbers generated by VAEs [4]. While derived features may work for a neural network, they can create noise in other methods. Finally VAEs may be weak to the same issues as neural networks, in this case causing an encoding with slight variance to result in a much different decoded value.

LORE, local rule-based explanations, [3] attempts to improve on local explanations by using rules. LORE generates perturbations using a genetic algorithm. Perturbations are produced by a mixture of "surviving" samples' features. Survival is determined by fitness in their case a function of the class similarity and distance (on a 0 to 1 scale) reduced by the point-wise similarity to the input. The approach is a good step forward since data are designed to be similar to input. The only concerns are on the likelihood that surviving samples are similarly distributed to the original data and are in-distribution. This points out a general disagreement behind what the end goal of a local explanation is since the establishing literature does not provide a clear outline. In LORE's case it is closer to a search problem that uses data as examples to search through. In this work we aim to clarify the goal, which may also reveal alternative solutions to the problem.

Another local explainer, black-box association-rule based explainer (BARBE) [5], also uses rules but through an associative classifier allowing complex combinations of model behaviour to determine outcome. An approach that has also been applied on non model-agnostic approaches, like explaining time series data [6]. The time-series approach does not generate perturbations, instead using learned, structured features. Clearly having more relevant perturbations should have an impact on performance, which we will investigate further in this work.

2.2. **Evaluation Criteria**

Fidelity is the main criterion of focus as it was popularized by LIME [1]. However, applications of fidelity have been limited, generally focusing on global fidelity (as informed by the training set used to produce perturbations) and local fidelity (as also used by [7] counterintuitively produced via "local" perturbed data) which is no more than a new label for the training accuracy. Some work uses all samples to produce explanations [2] and evaluates the final fidelity on these. Which may overshadow problems with how perturbations are produced. If there are more data on the mean of a distribution then many errors with producing perturbations will be corrected by their similarity to the mean. Finally being

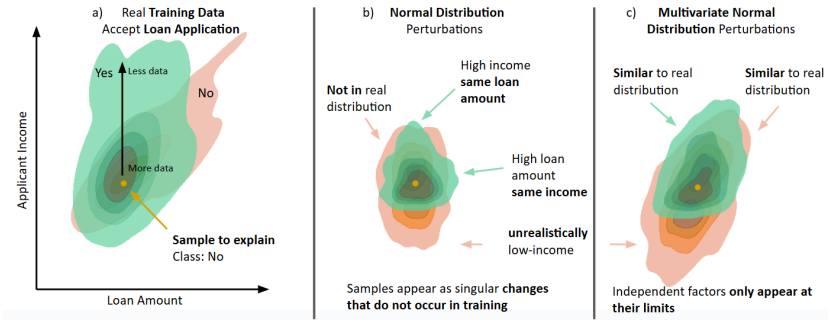


Figure 2. a) Distribution of real data on Applicant Income and Loan Amount (green "Yes", orange "No"). Four layers, by darkness of color, each are 20% of the data (outliers are not shown). b) Distribution of perturbed data about x_p (light orange dot, class "No") using a normal distribution. c) Perturbations using a scaled multivariate distribution (our change).

drowned out by the larger numbers of such samples. Genetic approaches like LOREs [3] use survival to produce only perturbations similar to the input value. Here the global fidelity is at least a comparison to unseen data. However the local fidelity is not a good evaluation of model performance, and a high global fidelity may inform us that the generated rule is generalizable rather than specific to the local instance. Approaches like BARBE use multiple rules [5], increasing the white-box complexity. Although performance may improve we need to verify that this is not from overfitting, when evaluating on perturbed data, and that the local rules retain high performance.

3. Problem Definition

Training and evaluation of surrogate-based explainers are limited to samples most local to the point being explained. LIME and VAE-LIME provide general overviews of their optimization criterion [1, 2], but there is sparse literature discussing what we expect the true structure being solved for looks like in a local explanation.

We must address three issues to evaluate how to improve performance measure: what may training look like; how are samples produced to train on; and what may a local explanation be, leading to how they may be evaluated.

3.1. Training a Local White-Box Explainer

Usually surrogate models are trained similarly to any other machine learning model. Here a white-box, $g : X \to [-1, 0) \cup (0, 1]$ (generally these are bounded on an interval $[0, 1]$ we include a center point zero to simplify some logic), trained on D_t will have ideal performance given a loss function $loss : \mathcal{R} \times \mathcal{R} \times \mathcal{R} \to \mathcal{R}_+$ to minimize, where the inputs are two sets of predictions and one set of weights for each.

As an example LIME uses a kernel weighting relative to an input sample x_p, with a kernel function $k : X \times X \to \mathcal{R}_+$, a local dataset D_p, and an evaluation of the complexity $\Omega : \mathcal{G} \to \mathcal{R}_+$ minimizing:

$$\underset{g \in \mathcal{G}}{argmin}(loss(f(D_p), g(D_p), k(D_p, x_p)) + \Omega(g))$$

We can stand to ignore using complexity, Ω since the focus here is on the final evaluation. The locality assumed of D_p and k will be revisited and addressed in our formal problem definition.

3.2. **Training on Perturbations**

We assume that data are distributed with a mean, μ, and a standard deviation, σ. Therefore our training data, D_t, may be represented approximately by $D_t = \{x \sim \mathcal{N}(\mu, \sigma^2)\}$. By replacing μ with input data instead x_p a set of training data local to x_p may be produced which we refer to as $D_p = \{x \sim \mathcal{N}(x_p, \sigma^2)\}$. To find σ, we may provide training data and derive the value for each feature or have user input parameters. An advantage we note of using perturbed samples is that these parameters may be provided since it is not necessarily the case that training data are available.

3.3. **Local Predictions from a Black-Box Problem Definition**

Given a black-box model $f : X \to [-1, 0) \cup (0, 1]$ we should be able to piece its behavior together using a combination of non-trivial functions $f^k : X \to [-1, 0) \cup (0, 1]$ of the same structures as f, where:

$$f(x) = d(x, \beta_1)f^1(x) + ... + d(x, \beta_n)f^n(x) \tag{3.1}$$

With a distance function $d : X \times X \to (0, 1]$ (near 0 when x is far from β_i and 1 being they are the same value) acting as weights from the reference points $\beta_i \in D_t$. In this case consider that we may reconstruct a kernel SVM by piecing together a series of local SVMs applied to different areas of the covered space (trained on different data). We may also piece together a one-layer neural network (which is interpretable) by applying each individual neuron's prediction and their given weights as a linear combination. There should then be some function f^{x_p} that represents the local behavior applied to x_p by f, if we remove the functions that satisfy the condition $-\epsilon \leq d(x_p, \beta_i)f^i(x_p) \leq \epsilon$, for some $\epsilon \in (0, 1)$, marking functions as either indeterminate on x_p (since the decision is not significantly on class -1 or 1) or too far away from it. There should then be a weighting of data w_x that can be used in conjunction with the full training dataset D_t to train just f^{x_p}. So during evaluation we can find an appropriate weighting to produce an evaluation criterion based on the behavior we expect on the local model applied to x_p. During optimization we may search for:

$$\text{ideally, } \underset{g \in \mathcal{G}}{argmin}(loss(f^{x_p}(D_t), g(D_t), \bar{d}(D_t, x_p))), \text{ we may use } \underset{g \in \mathcal{G}}{argmin}(loss(f(D_t), g(D_t), w_x)) \tag{3.2}$$

Either will results in a model best fit to the local functions that determine behavior. However, since f is a black-box model we do not have access to w_x or ways to solve it, as we may not have all the training data. We expect with black-boxes the local behavior functions cannot be solved for easily either (as it is a non-interpretable model), thus the ideal situation is also unreachable. So we instead rely on large sources of data in the local vicinity of x_p. As in Section 3.2 though there may be flaws which will be addressed in Section 4.3. Now we can rely on a simpler problem using a normal optimization as in many machine learning methods for our white-box model:

$$\underset{g \in \mathcal{G}}{argmin}(loss(f(D_p), g(D_p), 1))$$

Though not established for this purpose in the work on LIME the loss over kernel weights is supported by our assumptions about the format of a local model derived from the black-box. As discussed LIME uses perturbed data based on a simple normal distribution which may not produce data with realistic behavior. Something VAE-LIME addresses by training a VAE on nearest neighbors of x_p [2]. LORE instead weighs the survival of a newly generated sample according to its similarity to x_p [3], which may also help with the problem. BARBE aligns well with the potential structure to solve for in equation 3.1 given each independent rule or subset of rules can represent the pieces of black-box to solve for locally. Better

perturbations should make for better weights used to find these pieces. We consider only generating variations without weight but cross-validating for some locality by scaling down the variance (σ/η discussed in Section 4.3).

To see if it is needed in optimization we use local fidelity (to account for the desired evaluation eq. 3.2) for evaluation. We sub in for w_x a distance function d in place of the kernel function of LIME. We cautiously assume that d is a reasonable approximation of the true distance function and we can use x_p as an anchor. The given black-box f tends to not make predictions of 0 on points far from x_p (since it exhibits behaviors from every local function). We expect using weighted fidelity during training may cause the white-box to learn a global model if there is bias for data to be far from x_p. Also, if weights were used with local data it may be redundant since the bias is centered on x_p anyways. We still expect local fidelity to have a role in building user trust since while a local explanation will perform poorly on all the data it should perform well on the local subset.

4. **Our Approach**

We expect that for post-hoc methods aimed at determining specifics of decision making rather than generic feature importance the quality of perturbed data will be a limiting factor. With current evaluation metrics we cannot accurately evaluate this assumption, since when considering only perturbation fidelity over-fitting is always the best approach. We also acknowledge the out of distribution problem when using perturbed data, which our approach aims to limit.

In Section 2 we acknowledged potential issues from modern perturbation approaches being unexplainable. While finding a goal in Section 3.3 we also acknowledged limitations to available weights that may not entirely be accounted for by distributions and prior techniques. We propose solutions to these two problems; first we address the weight limitations by proposing simple weighting techniques evaluated on real data, so an overfit model will not appear to have perfect performance as a result of limited perturbations. These tests are referred to as single-blind and double-blind. Based on whether the black-box was trained on the data used to calculate fidelity. We experiment with two distance measures based on Euclidean distance and Nearest Neighbors and propose scaling back perturbations using a general framework which can include several different distributions. All distribution parameters can be user provided and have the added advantage of being debuggable so an otherwise clear explainer is not rendered useless by complex, untraceable perturbation choices. The full process and changes are summarized in figure 1 and shared on GitHub, linked in Appendix E. The remainder of this section goes into detail about these approaches.

4.1. **Single-Blind and Double-Blind Fidelity**

$$fidelity(f, g, X, w) = \frac{\sum_{k=0}^{|X|} w_k \mathbb{I}(f(X_k) = g(X_k))}{\sum_{k=0}^{|X|} w_k} \tag{4.1}$$

We note that in Table 1 (lines 1, 2, and 3 of Fidelity) there is a disconnect between performance on perturbations and real data. So a local real data example may be integral to trusting the explanation. At least for experimental discussion, even if during application we do not have access to this data. Even if new data is used for original "local" fidelity measures (eq. 4.1 with $w = 1$) it will not address if the perturbations are common or rare in the data, only if the potentially spare area around x_p was learned.

We propose Single and double-blind tests, named as such since one model has previous access to the data while the other does not. In the case of single-blind tests this is training data, which we expect there may be some of remaining from initial training of a model. The

aim of using this data for fidelity is that we can assess performance to the ideal performance the model should achieve (from Section 3.3). In comparison the double-blind data similar in purpose to initial tests is to see how the white-box will generalize to cases that were not used to produce perturbations and how well it truly represents the black-boxes behavior. In many cases the double-blind test may be the only data we have access to, so we recommend keeping a hold out set to test on and training perturbations with another set, serving a similar purpose to single-blind tests just only for the black-box.

4.2. **Local Weights**

In Section 3.3 we theorized that local functions are linked to a distance determined by local anchors. In evaluation we want to apply this idea one simple option we consider is the Euclidean distance:

$$d(x) = (\sum_{n}^{i=0} |x_i - x_{p,i}|)^{1/2}$$

$$w_{euclidean}(x) = \frac{\sum_{l}^{k=0} d(X_k)}{d(x) + \epsilon} \tag{4.2}$$

Where $\epsilon = 1e - 6$ to represent a small value without dividing by zero. We can also use a nearest neighbors approach, which better accommodates for a lack of certainty about the distance function since only the nearest neighbors, which should have the same local model applied, are compared with:

$$q = quartile(d(x), \{20, 100\}),\ w_{nn}(q) = \frac{contribution(q)}{\sum_{k=0}^{|X|} \mathbb{I}(q_k = q)} \tag{4.3}$$

Where $contribution(q)$ is a dictionary of values mapping quartiles to their total weight, in this case we use $\{20 = 100, 100 = 0\}$ describing in this example that the closest 20% of the data is 100% of the accuracy. In other words if we have 200 samples the nearest 40 neighbors to x_p will be used equally in evaluation and the remainder of data will be ignored. The method does allow for other quartiles to be used at different weights, for a more complex approach to comparison.

4.3. **Scaled and Alternative Perturbations**

$$D_p = \{x \sim \mathcal{N}(x_p, (\sigma/\eta)^2)\} \tag{4.4}$$

Making x_p the mean and adding a term $\eta \in [1, \infty)$ as a factor that scales down the deviation being applied to the noise simulated by reducing the variance in the distribution $(\sigma/\eta)^2$. To keep the perturbations local, otherwise the distribution would simulate data similar to the entire dataset, encouraging a global explanation. Finally a more advanced representation of the data is the multivariate normal distribution:

$$D_p = \{x \sim \mathcal{N}(x_p, \Sigma/\eta)\} \tag{4.5}$$

Where Σ is the covariance of the data, estimated from D_t or manually entered by a user. Covariance includes the original standard deviations, σ, along the diagonal and covariance values $cov_{ij} = cov_{ji}$ (which may be derived using correlation coefficients, $c_{ij} = c_{ji} \in [-1, 1]$, where $i \neq j$ representing relationships between the features i and j). The multivariate normal better represents situations that may actually occur in a dataset rather than independent effects as displayed in Figure 2.

4.4. **Perturbing Encoded Categorical Variables**

Perturbing encoded categorical values of a sample equivalently to the training data would cover many combinations of categories, with a high cost. We aim to produce a local explanation, so some amount of bias towards current categories is justified. We change the encoded values so $x_p^j = \bar{x}^j$, the mean value of the one-hot value. We then provide a bias to the current value, $x_p^j = \bar{x}^{\neg j} + \bar{x}^j$, where $\bar{x}^{\neg j} = (\sum_{i \in [i, i+n] \neg j} x^i)/n$, the average of the other categories. To standardize the odds we also set the deviation values to a constant $\sigma = 0.3334$. We use this value since in the case of a normal distribution we expect the maximum to be $\pm 3\sigma \approx \pm 1$ deviations. In combination with the bias this means two equally seen categories would have the means $\mu_1 = 0.5$, $\mu_2 = 0.25 + 0.5$ and the category would change frequently with a small amount of bias towards remaining the same (only when $(\epsilon_2 - \epsilon_1)\sigma \leq -0.25$ or a shift totaling -0.25 for feature 2 odds is observed). We adjust the covariance values accordingly and end up with a skewed odd that the feature remains the same while allowing for correlations between certain values to have a strong effect on outcome. The user need only input approximate odds of each category and some expected correlations. The final data are output in their original format by applying equations C.1.

Pert. Method	Eval. Data	Fidelity	Euclidean F.	Nearest N. F.	Hit Rate
Normal	Perturbed	**0.606**±0.082	N/A	N/A	0.467
	Single-B	0.177 ± 0.078	0.253 ± 0.078	0.376 ± 0.152	
	Double-B	0.168 ± 0.089	0.227 ± 0.099	0.329 ± 0.227	
Mult-Normal	Perturbed	0.573 ± 0.088	N/A	N/A	**0.833**
	Single-B	**0.288**±0.128	**0.453**±0.243	**0.577**±0.210	
	Double-B	**0.312**±0.144	**0.444**±0.204	**0.637**±0.274	
t-Distribution	Perturbed	0.560 ± 0.079	N/A	N/A	0.467
	Single-B	0.178 ± 0.079	0.244 ± 0.061	0.357 ± 0.121	
	Double-B	0.170 ± 0.067	0.227 ± 0.063	0.295 ± 0.186	
Uniform	Perturbed	0.584 ± 0.046	N/A	N/A	0.4
	Single-B	0.179 ± 0.056	0.247 ± 0.059	0.363 ± 0.141	
	Double-B	0.150 ± 0.058	0.193 ± 0.056	0.217 ± 0.159	
Cauchy	Perturbed	0.212 ± 0.090	N/A	N/A	**1**†
	Single-B	0.135 ± 0.075	0.291 ± 0.267	0.327 ± 0.237	
	Double-B	0.144 ± 0.111	0.240 ± 0.214	0.307 ± 0.264	

Table 1. LIME+P performance (percentage 100% = 1.000) with different distributions explaining a Random Forest black-box trained on the UCI Libras dataset. Normal distribution is the same as LIME, used here as a baseline. **Bold** values represent the best or near-best performance among the methods for each performance measure. N/A in weighted perturbed fidelity, as all data is assumed to be local. † Cauchy's high hit rate likely a result of only producing two of fifteen classes (single-blind fidelity $\approx 2 * 0.0667$).

5. **Experimental Setup**

Experiments are conducted by randomly selecting between 20 and 50 samples from each dataset to explain. The remaining data are then split into training (single-blind) and testing (double-blind) sets, with an 80-20 split. The training data are used to train a black-box model from among random forests (5 decision trees with a max depth of 4) and neural networks (ReLu decision with 2 hidden layers (100, 50)). A 1/3 proportion of the training data are also provided to each explainer to find settings for perturbations to simulate limited access to training data or uncertain parameters.

A baseline example for each test and evaluations are conducted using the unchanged local normal distribution that LIME uses. VAE-LIME's and LORE's perturbations and strategies are compared to in-terms of success, but are not provided to each model. Each approach

including ours will be trained on 1000 perturbations as a standard, and usually default, number of perturbations. We cross-validate on the single-blind test to select explainer settings cross-validation is described in details in Appendix A to compare using the double-blind evaluation. We also evaluate models using hit rate [2]. Hit rate for each dataset is the average number of times the model correctly classified the sample to perturb, the successful "hits" being evaluated further.

5.1. **White-Box Models**

We consider ridge regression (LIME [1]), decision tree (LORE [3]), and association rule (BARBE [5]) models as explainers. LIME, VAE-LIME [2], and LORE additionally provide comparisons to the perturbation method we apply. BARBE and LORE use surrogate models that give rules based on support in the data. Better data should lead to more accurate and relevant rules giving more utlity to the approach.

We use pre-made implementations of each method when they are available on GitHub[1][2][3] except for BARBE where we use a privately shared version, and have since made it available on GitHub[4]. Implementations of LIME, VAE-LIME, and LORE are superficially modified to provide the perturbed data and trained models in an accessible format to conduct tests on. Predictions from LIME's multiple ridge regressions are simplified to:

$$c = \underset{c \in C}{argmax}(f^c(x))$$

Where f^c is the predictor trained on data for class c. Used as the prediction from LIME's surrogate. We do not expect these modifications to have a significant impact on the explanations provided by these techniques. As a classifier the application of ridge regressions here is expected to be the same. We do modify the input to LIME to better compare with other techniques and make experiments similar to the application environment. LIME would typically take $y = \{p_1(x), p_2(x), ..., p_n(x)\}$ instead of the full range of probabilities we set $p_i(x) = 0$, if $f(x) \neq i$, and $p_k(x) = 1$, if $f(x) = k$.

5.2. **Choice of Distributions**

To cover a large range of possible underlying distributions we choose from among five. Some that function as in equation 4.4 (fig. 2 b). The normal distribution as it is the approach used in LIME, although we add optional reductions to σ values, scaling them to be local. We apply the multivariate normal distribution (eq. 4.5, fig. 2 c). The t-distribution since it can accommodate for slightly more skewed data, and in our case can provide some outlier cases to training. The uniform distribution since it is usually used to produce consistent noise over a sample, noise that may be easier to train on. Finally, the Cauchy distribution as it produces far outliers but with high uniformity in the tails, it may be a good mixture of a normal and uniform distribution. All distributions are used in their implementation via the SciPy package [8].

5.3. **Datasets**

We use four datasets: the UCI Iris dataset [9], the UCI Diagnostic Breast Cancer dataset [10], Loan Acceptance[5], and the UCI Libras Hand Movement dataset [11]. The Iris dataset

[1] https://github.com/marcotcr/lime
[2] Exact implementation is not available so a similar one is used: https://github.com/domenVres/Robust-LIME-SHAP-and-IME
[3] https://github.com/riccotti/LORE
[4] https://github.com/amiixai/barbe
[5] https://www.kaggle.com/code/yonatanrabinovich/loan-prediction-dataset-ml-project/notebook

is commonly used as a baseline since it only has 4 geometric features and three classes, which are separable into clusters. Geometric features lead to improper perturbations producing invalid negatives, which our new tests would identify if they risk classification usefulness. The UCI Breast cancer dataset is composed of 30 numerical features, only one third of the training data are used for finding distribution parameters creating potential issues for the covariance values to be accurately found. The Loan dataset include a mixture of categorical variables our approach aims at improving classification on. Although the data are 10 features, with an abundance of categories the one-hot encoding turns this into more features. Making the loan data difficult to represent well with perturbations. The UCI Libras data is selected since time-series data have structured correlations that are important to making predictions, it also offers the challenge of recognizing 15 classes. Accounting for these correlations our method should produce more viable structured data to learn from, rather than producing noise that may result in a white-box over-fitting to the perturbations. Due to the size of these data we only use every other feature, reducing the total from 90 to 45 and run only tests for $\eta = 1$.

Data	Pert.	LIME+P	BARBE+P	LIME	BARBE	LORE	VAE-LIME
Iris	M-Norm.	$\mathbf{0.86}^{\ddagger}\pm0.17$	0.65 ± 0.18	$\mathbf{0.85}^{\ddagger}\pm0.13$	$\mathbf{0.74}\pm0.15$	0.67 ± 0.21	$0.54^{\ddagger} \pm 0.17$
	t-Dist.	$0.84^{\ddagger} \pm 0.13$	$\mathbf{0.74}\pm0.18$				
	Uniform	$0.84^{\ddagger} \pm 0.11$	0.70 ± 0.20				
	Cauchy	$0.78^{\ddagger} \pm 0.13$	0.71 ± 0.18				
BC	M-Norm.	$\mathbf{0.91}^{\ddagger}\pm0.13$	$\mathbf{0.86}\pm0.11$	$0.27^{\ddagger} \pm 0.37$	0.42 ± 0.27	0.56 ± 0.22	$0.80^{\dagger} \pm 0.14$
	t-Dist.	$0.18^{\ddagger} \pm 0.33$	0.41 ± 0.28				
	Uniform	$0.23^{\ddagger} \pm 0.33$	0.39 ± 0.31				
	Cauchy	$0.44^{\ddagger} \pm 0.34$	0.33 ± 0.31				
Loan	M-Norm.	$0.45^{\dagger} \pm 0.22$	$\mathbf{0.63}\pm0.17$	$0.50^{\dagger} \pm 0.18$	$\mathbf{0.66}^{*} \pm 0.18$	$0.60^{*} \pm 0.21$	$\mathbf{0.66}^{\dagger}\pm0.17$
	t-Dist.	$0.42^{\dagger} \pm 0.24$	0.55 ± 0.20				
	Uniform	$0.39^{\dagger} \pm 0.21$	0.60 ± 0.21				
	Cauchy	$0.48^{\dagger} \pm 0.18$	0.58 ± 0.17				
Libras	M-Norm.	$0.32^{\ddagger} \pm 0.32$	$\mathbf{0.29}\pm0.19$	$0.25^{\ddagger} \pm 0.22$	0.18 ± 0.14	$\mathbf{0.30^{*}}\pm0.21$	$0.40^{\dagger} \pm 0.24$
	t-Dist.	$0.27^{\ddagger} \pm 0.25$	0.25 ± 0.16				
	Uniform	$0.27^{\ddagger} \pm 0.35$	0.24 ± 0.17				
	Cauchy	$0.26^{*} \pm 0.22$	0.11 ± 0.11				
Avg.	Best	$\mathbf{0.643}^{\ddagger}$	$\mathbf{0.630}$	$0.468^{\ddagger}$	0.483	0.533	$0.600^{\ddagger}$
	M-Norm.	$0.635^{\ddagger}$	0.608				
	t-Dist.	$0.428^{\ddagger}$	0.488				
	Uniform	$0.433^{\ddagger}$	0.483				
	Cauchy	$0.490^{\ddagger}$	0.433				

Table 2. Nearest neighbors fidelity of double-blind test performance with different distributions and datasets explaining a Neural Network black-box. Explainer settings chosen by cross-validating with respect to the single-blind performance. *†‡ Low hit rates ($\ddagger$<0.334, $\dagger$<0.667, and $*$<0.95).

6. **Results**

We note the low hit rate of LIME in general as a potential risk for performance. We noticed that LIME failed more frequently for complex models, as the results from table 1 conducted on the random forest black-box has much higher success than any of the results for the neural network (Hit Rate > 0.6667). The data here are from the hardest case (Libras with 15 classes) but also the one we expect to see interactions between variables help with the most. Given the high Hit Rate it is not surprising that the overall performance is better in each evaluation criterion for the approach using multivariate normal distributions. We note that if selections were done using just fidelity on perturbed data, as is often referred to as local fidelity in other works, a much worse method would have been selected.

We next aim to determine the best general approach for both LIME and BARBE, compared to other perturbation approaches from LORE and VAE-LIME. The results in table 2, from a neural network black-box, show that in general the multivariate normal improves performance on the double-blind fidelity tests. In most cases for LIME it also improves Hit Rate as with our initial observations, BARBE already had a high hit rate and this approach improved the accuracy. As expected the Libras dataset is the hardest problem for LIME and although double-blind fidelity improved using multivariate normals to perturb the data hit rate actually decreased. We expect this may have been due to using a complex neural network. So while the data provided a better representation of the space this complicated learning. The simple model that LIME uses cannot accommodate and end up failing. This conclusion is also supported by the success of BARBE, specifically when using the more complex dataset.

Data	Eval.	LIME+P	BARBE+P	LIME	BARBE	LORE	VAE-LIME
Iris	Pert. Fid.	0.87 ± 0.06	0.84 ± 0.07	0.87 ± 0.04	0.83 ± 0.03	**0.99**$\pm$0.01	0.97 ± 0.06
	SB Fid.	**0.83**$^{\dagger}\pm$0.10	0.73 ± 0.14	0.76 ± 0.16	**0.78**$\pm$0.14	0.73 ± 0.15	0.58 ± 0.19
	DB Fid.	**0.79**$^{\dagger}\pm$0.17	0.69 ± 0.16	0.74 ± 0.16	**0.74**$\pm$0.19	0.72 ± 0.16	0.49 ± 0.20
	Hit Rate	0.65	**1**	0.633	**1**	**1**	0.533
BC	Pert. Fid.	0.92 ± 0.05	**0.99**$\pm$0.01	0.86 ± 0.07	0.99 ± 0.02	0.98 ± 0.02	0.97 ± 0.02
	SB Fid.	**0.92**$^{*}\pm$0.08	**0.89**$\pm$0.08	0.60 ± 0.27	0.67 ± 0.20	0.74 ± 0.17	0.87 ± 0.09
	DB Fid.	**0.93**$^{*}\pm$0.09	**0.89**$\pm$0.01	0.60 ± 0.27	0.67 ± 0.20	0.74 ± 0.17	0.86 ± 0.11
	Hit Rate	0.7	**1**	0.72	**1**	**1**	0.61
Loan	Pert. Fid.	0.72 ± 0.05	0.87 ± 0.04	0.73 ± 0.05	0.83 ± 0.04	**0.97**$\pm$0.08	0.85 ± 0.06
	SB Fid.	0.61 ± 0.20	**0.81**$\pm$0.10	0.64 ± 0.19	0.76 ± 0.19	0.78 ± 0.07	0.76 ± 0.11
	DB Fid.	0.65 ± 0.23	**0.81**$\pm$0.13	0.68 ± 0.20	0.76 ± 0.20	0.79 ± 0.16	$0.80^{\dagger} \pm 0.13$
	Hit Rate	0.68	0.98	0.69	0.94	0.95	0.64
Libras	Pert. Fid.	0.58 ± 0.09	0.79 ± 0.04	0.70 ± 0.11	**0.99**$\pm$0.00	0.93 ± 0.02	0.84 ± 0.09
	l-SB Fid.	**0.48**$^{\dagger}\pm$0.22	**0.43**$\pm$0.16	0.38 ± 0.18	0.23 ± 0.11	0.41 ± 0.19	0.39 ± 0.20
	l-DB Fid.	**0.48**$^{\dagger}\pm$0.30	**0.42**$\pm$0.22	0.29 ± 0.06	0.18 ± 0.15	0.42 ± 0.24	**0.47**$^{\dagger}\pm$0.24
	Hit Rate	0.5	**1**	0.417	**1**	0.8	0.417
Avg.	Pert. Fid.	0.773	0.873	0.790	0.910	**0.968**	0.908
	l-SB Fid.	0.710	**0.715**	0.595	0.610	0.665	0.650
	l-DB Fid.	**0.713**†	**0.703**	0.578	0.588	0.668	0.655
	Hit Rate	0.633	**0.995**	0.615	0.985	0.938	0.550

Table 3. Performance metrics compared across the different approaches averaged for Random Forest and Neural Network black-boxes for settings selected via cross validation, except for Libras where only distribution is selected.

Our final comparison had to do with general performance of our chosen approach (multivariate normal distributions) against each other method for both tests settings, random forest and neural network. The results in table 3 show that on average BARBE with multivariate perturbations performs the best. We find that on average LIME sees a performance improvement, typically also benefiting from a higher hit rate with multivariate distributions. The approach, of adding bias to categorical data, is sound since it does not reduce performance, but it also does not appear to improve performance much, more testing on datasets that combine categorical and numerical data where we expect the best improvements to be seen.

7. **Conclusion**

We found that the complex, but still understandable multivariate normal distribution produced perturbations that routinely trained explainers better aligned to the black-box in LIME and BARBE. Though the performance of LIME was low, better perturbations still improved it. We also saw in more complex explainers, like BARBE, the improved

perturbations propagated to overall better performance. Multivariate normal distributions are still feasibly generated by user input and tend to produce data that is more difficult to train on for BARBE, avoiding overfitting. We found that other measures for local fidelity (evaluated on perturbed data) were misleading, with little correlation between perturbed fidelity and performance on our tests. Leading us to conclude that weighted Single-Blind and Double-Blind tests serve an integral role in proving the best performer among post-hoc explainers.

Acknowledgments

The authors gratefully acknowledge funding from the Natural Sciences and Engineering Research Council (NSERC) of Canada and the Canada CIFAR AI Chairs Program for Amii. The authors also acknowledge the thoughtful recommendations and discussions with the Amii XAI Lab at the University of Alberta which helped to motivate this work.

References

[1] M. T. Ribeiro, S. Singh, and C. Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *CoRR* abs/1602.04938 (2016). arXiv: 1602.04938. URL: http://arxiv.org/abs/1602.04938.

[2] X. Xiang, H. Yu, Y. Wang, and G. Wang. "Stable local interpretable model-agnostic explanations based on a variational autoencoder". In: *Applied Intelligence* 53.23 (Sept. 2023), 28226–28240. ISSN: 0924-669X. DOI: 10.1007/s10489-023-04942-5. URL: https://doi.org/10.1007/s10489-023-04942-5.

[3] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, and F. Giannotti. "Local Rule-Based Explanations of Black Box Decision Systems". In: *ArXiv* abs/1805.10820 (2018). URL: https://api.semanticscholar.org/CorpusID:44063479.

[4] U. Garay-Maestre, A.-J. Gallego, and J. Calvo-Zaragoza. "Data Augmentation via Variational Auto-Encoders". In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Ed. by R. Vera-Rodriguez, J. Fierrez, and A. Morales. Cham: Springer International Publishing, 2019, pp. 29–37. ISBN: 978-3-030-13469-3.

[5] M. Motallebi, M. T. A. Anik, and O. R. Zaïane. "Explaining decisions of black-box models using barbe". In: *International Conference on Database and Expert Systems Applications*. Springer. 2023, pp. 82–97.

[6] O. Bahri, P. Li, S. F. Boubrahimi, and S. M. Hamdi. "Temporal rule-based counterfactual explanations for multivariate time series". In: *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2022, pp. 1244–1249.

[7] P. Rasouli and I. C. Yu. "Meaningful Data Sampling for a Faithful Local Explanation Method". In: *Intelligent Data Engineering and Automated Learning – IDEAL 2019: 20th International Conference, Manchester, UK, November 14–16, 2019, Proceedings, Part I*. Manchester, United Kingdom: Springer-Verlag, 2019, 28–38. ISBN: 978-3-030-33606-6. DOI: 10.1007/978-3-030-33607-3_4. URL: https://doi.org/10.1007/978-3-030-33607-3_4.

[8] P. Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[9] R. A. Fisher. *Iris*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C56C76. 1936.

[10] M. O. S. N. Wolberg William and W. Street. *Breast Cancer Wisconsin (Diagnostic)*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5DW2B. 1993.

[11] P. S. Dias Daniel and H. Bscaro. *Libras Movement*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5GC82. 2009.

Appendix A. Explainer Cross-Validation Settings and Black-Box Choices

Cross-validation setting for the new approach select from $\eta = \{1, 2, 3, 5, 10\}$ and pert_dist = {'normal' (baseline), 'multivariate-normal', 't-distribution', 'uniform', 'cauchy'}. Evaluations using standard fidelity, Euclidean weighted fidelity (eq. 4.2), and Nearest Neighbors fidelity (eq. 4.3) are used. We use small steps of η since by $\eta = 3$ outlying values are what was most common in the original. With high η values $\{5, 10\}$ serving as extreme cases.

The black-box models are chosen to offer two difficulties of explanation and so no white-box has only preferable black-boxes (e.g. decision tree explaining random forests and ridge regressions (sigmoid) explaining neural networks).

Appendix B. Possible Limitations

In Section 2 we discussed that some methods treat learning more like a search task. The approach is repeated in other work, with the justification of producing samples that do have different classes. Though in many cases these are unexplainable we acknowledge that without different classes learning is not possible. The current framework does not guarantee that there is at least one sample of a different class from x_p which may lead to difficulty learning.

Although we account for correlations during perturbation (consider again Figure 2 (b) vs (c)) in this approach there may be different behaviours within groups in the data. As a result, the local behaviour would not accurately be represented, it would be improperly characterized by the global behaviour. We see this in the Iris results for Table 2. In Figure 2 we note unrealistic behaviour in (b) and (c). Scaling can help with unrealistic low income that we note both cases have. However, bounding may be a better solution we aim to implement in future work.

Appendix C. Encoding Categorical Variables

Categorical data is one-hot encoded into a vector. For example in a variable which can take the values "yes", "no", and "maybe" the value for "maybe" and "yes" may be encoded as (001) and (100) respectively. Assuming we have some series of one-hot encoded values x^i to x^{i+n} we may return them to the associated value in a list of values as:

$$x^k = key(k, \underset{j \in [i, i+n]}{argmax}(x^j)) \tag{C.1}$$

Where $key(k, j)$ is a function that gets the discrete value encoded at the j^{th} position for variable k. As derived from the original categories encoded by:

$$\forall j \in [i, i+n],\ x^j = \mathbb{I}(x^k = key(k, j)) \tag{C.2}$$

Appendix D. Adjusting Covariance Values for Biased Categorical Values

As we made changes to the variance for categorical values in Section 4.4 there are changes to related covariance values. The shift has no impact, however the variance change $\sigma = 0.334$ does. The change is relatively simple, as $cov_{i,j} = c_{ij}\sigma_i\sigma_j$ we need only insert the new sigma through multiplying a term $\sigma'_i = (0.334)/\sigma_i$. Resulting in $cov'_{i,j} = cov_{i,j}\sigma'_i = c_{i,j}(0.334)\sigma_j$.

Appendix E. Source Code

Implementations written as wrappers for LIME, LORE, and VAE-LIME and a full implementation of the changes implemented on BARBE available on GitHub https://github.com/amiixai/barbe.