Hindawi
International Journal of Intelligent Systems
Volume 2023, Article ID 8068569, 34 pages
https://doi.org/10.1155/2023/8068569

WILEY | Hindawi

Research Article

REVEL Framework to Measure Local Linear Explanations for Black-Box Models: Deep Learning Image Classification Case Study

Iván Sevillano-García, Julián Luengo, and Francisco Herrera

Department of Computer Science and Artificial Intelligence, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada, Granada 18071, Spain

Correspondence should be addressed to Iván Sevillano-García; isevillano@ugr.es

Received 15 November 2022; Revised 18 April 2023; Accepted 8 May 2023; Published 3 June 2023

Academic Editor: Alexander Hošovský

Explainable artificial intelligence is proposed to provide explanations for reasoning performed by artificial intelligence. There is no consensus on how to evaluate the quality of these explanations, since even the definition of explanation itself is not clear in the literature. In particular, for the widely known local linear explanations, there are qualitative proposals for the evaluation of explanations, although they suffer from theoretical inconsistencies. The case of image is even more problematic, where a visual explanation seems to explain a decision while detecting edges is what it really does. There are a large number of metrics in the literature specialized in quantitatively measuring different qualitative aspects, so we should be able to develop metrics capable of measuring in a robust and correct way the desirable aspects of the explanations. Some previous papers have attempted to develop new measures for this purpose. However, these measures suffer from lack of objectivity or lack of mathematical consistency, such as saturation or lack of smoothness. In this paper, we propose a procedure called REVEL to evaluate different aspects concerning the quality of explanations with a theoretically coherent development which do not have the problems of the previous measures. This procedure has several advances in the state of the art: it standardizes the concepts of explanation and develops a series of metrics not only to be able to compare between them but also to obtain absolute information regarding the explanation itself. The experiments have been carried out on four image datasets as benchmark where we show REVEL's descriptive and analytical power.

1. Introduction

In recent years, artificial intelligence (AI) has experienced a huge development, providing solutions to many real-life problems. Unfortunately, these systems remain characteristically opaque, which is known as the black-box problem. To tackle the comprehension of the black box, several explainable AI (XAI) techniques have been proposed [1]. In general, the aim is to extract knowledge from black-box models so that they become understandable by a human but it also aims to show the risks of not using the XAI perspective [2].

In the literature, there is a clear separation between model-agnostic and model-specific explanations. Explanations designed as agnostic do not require knowledge of the model's own structure information [3, 4]. One of the most used and simple ones is local linear explanation (LLE).

All proposed explanations are based on different notions of what constitute an explanation and, therefore, are not directly comparable. In the literature, there are several proposals to compare explanations. In [5], different desirable qualitative aspects for an explanation are proposed, without including ways to measure them. In [6], the LEAF framework is proposed, designed for the evaluation and comparison of explanations. This framework has 4 different metrics to evaluate different desirable qualitative aspects of explanations. However, these metrics have different design inconsistencies which make them incomplete and biased.

Although there are different measurement proposals, there is no consensus in the XAI literature on how to evaluate explanations since there is no definition of what constitutes a good explanation [7]. Moreover, these measures have theoretical inconsistencies, and although they are useful to compare explanations, they do not provide absolute information on the explanation itself. Therefore, a set of robust metrics theoretically correct and representing characteristic behaviors of the method in practice is necessary. We also want to emphasize the difficulty of analyzing different factors that must inherently modify the explanation, such as the specific task covered by an AI or the type of data on which the explanation is generated.

Although there is no consensus within the literature on how we should create or even measure explanations, there are different state-of-the-art tools available that, combined with robust mathematical development, can provide a more generalizable and reliable analysis of the black-box generated explanations.

This work focuses on the proposal of the REVEL framework (Robust Evaluation VEctorized Local-linear-explanation), whose main contribution is to offer a consistent and theoretically robust analysis of the black-box generated explanations, as well as being useful at a practical level for the evaluation of explanations. REVEL takes advantage of the existing state of the art and develops a series of theoretical improvements on the generation and evaluation methods. In addition, it redefines and proposes different quantitative measures to robustly assess different qualitative aspects of the explanations. These measures emerge naturally and are well defined, so that we can extract not only comparative information among explanations but also get an absolute idea about the quality of an explanation on its own.

Although the theoretical study is generalizable to any kind of data and any kind of task, we focus on image classification in order to simplify the final discussion of the article. In addition, it is easier to work with images for the purpose of the analysis in the article, since it is simpler to generate different number of features with this data type.

The experimental section has been designed to show the analytical and descriptive potential of REVEL. We have designed three different scenarios on which to use REVEL. These scenarios are as follows:

(i) We analyze within LIME how much the number of black-box evaluations affects the quality of the explanations.

(ii) Within LIME, we also analyze how the number of features in which we split an image is affecting.

(iii) We compare the two well-known state-of-the-art black-box explanation generators, LIME and SHAP, to demonstrate the comparative capability of REVEL.

The rest of the paper is organised as follows. Section 2 provides a survey of motivations and basic concepts of LLE and describes two main methods that we will compare, LIME and SHAP. Section 3 proposes REVEL framework and highlights its strengths with respect to other methods of evaluating explanations in a theoretical way. Section 4 develops a generic experimental pipeline for the comparison of explanations which we use in Section 5 to perform a comparison of different aspects of LIME and SHAP on four image classification benchmarks. Finally, the concluding remarks and future work are reported in Section 6.

This paper is based on a preprint version published in [8].

2. Preliminaries: Considerations to Generate Local Linear Explanations

In this section, we review the type of explanations named LLE, also called feature importance models, additive feature attribution methods, or linear proxy models. These methods are called LLE because they are a local linear approximation of the black box. We focus on LLEs because of their rigor and simplicity, which helps when developing possible metrics. Other more complex explanations would make this task more difficult.

This section starts with a theoretical description of LLEs and describes the two state-of-the-art LLEs, LIME and SHAP. We then discuss four fundamental aspects for the generation of feature importance explanations: the differences between the concept of importance and how to compare them and how to generate the neighborhood of examples for the regression of LLEs and different considerations about the type of data we work on and the specific task we tackle.

2.1. Local Linear Explanations. Formally, let $X \subset \mathbb{R}^F$ be the input dataset. Let $f: \mathbb{R}^F \longrightarrow \mathbb{R}^C$ be the original black-box model, where C is the dimension of the output space $\mathcal{Y}$. Previous works define f as a function that relies on just $\mathbb{R}$, but in case of tasks such as nonbinary classification problems, the model output is a vector of probabilities where each component depends on all others. Let $x \in X$ be the input to be explained. A white-box LLE explainer is a function $g: \mathbb{R}^F \longrightarrow \mathbb{R}^C$ defined as follows:

$$g(x) = Ax + B, A \in \mathcal{M}_{F,C}, B \in \mathbb{R}^C, \tag{1}$$

and in other words, g is a linear application from the feature space to the output space.

Intuitively, the weights of both A and B are linked to the importance of each feature. More precisely, each weight $a_{i,j}$ of matrix A is linked to the importance of feature i to output j. Also, each bias b_j is linked to the general importance of output j.

The different LLE methods use linear regression minimizing error as follows:

$$\mathcal{L}(f, g, \pi_x) = \sum_{z \in N(x)} \pi_x(z)(f(z) - g(z))^2, \tag{2}$$

where the weight function selection depends on each particular method. Another factor to consider is how the neighbors are sampled. The original proposals consider a Bernoulli experiment for each feature, that is, each feature has the same probability to be present on the generated

neighbor. On the other hand, there are other newer proposals that consider a smart perturbation generation [9], where examples that contribute more to the explainability white-box model are more likely to be generated. For each LLE method, we use the sample-wise approach.

2.2. Models of Local Linear Explanations: LIME and SHAP. Once explained what LLEs are, we are going to describe the two main state-of-the-art LLEs, linear model-agnostic explanation (LIME) and Shapley additive explanation (SHAP). Although both are LLEs, they have clear differences in performing the black-box regression. We now describe how each method works and the main differences between them.

2.2.1. LIME. The LIME method [10] adopts the concept of local importance, which means that a feature that produces significant changes in the neighborhood of x is very important. Therefore, features that are important for the classification of x but do not produce significant changes in the neighborhood of x will end up being discarded as an important feature.

Formally, LIME builds a LLE model g by linear regression over a neighborhood $N(x)$ of the original datapoint x. The definition of this neighborhood is not trivial due to each dataset's different nature. In order to find a LLE g, LIME fits a ridge regression to $N(x)$ with the linear least squares function with the default kernel:

$$\pi_x(z) = \exp\left(\frac{-d(x,z)^2}{\sigma^2}\right), \tag{3}$$

where $d(\cdot,\cdot)$ is the euclidean distance and σ is a regularization factor.

The generation of the neighborhood $N(x)$ is performed by sampling from an exponential distribution with $\lambda = 1/\sigma$ a value v' with σ, the parameter selected for the LIME kernel. Finally, let $v = \lfloor v' \rfloor$. In the hypothetical case of $v > F + 1$, $v = F$ where F is the number of all features. The value v sampled is used to select randomly v features to exclude on this sample.

2.2.2. SHAP. The SHAP method [11] considers a feature to be important for the classification of an example x if it produces significant changes when compared to background values.

Formally, SHAP builds a LLE model g by computing the contribution of each feature to the prediction from a game theory approximation. This method tries to find a LLE g as a regression with the following kernel function, which is the SHAP kernel π_x defined as follows:

$$\pi_x(z) = \frac{F-1}{\binom{F}{|z|}(F-|z|)|z|}, \tag{4}$$

where $z \in \{0,1\}^F$ is a binary vector representing the presence of each of the F features on the z example and $\binom{N}{M}$ is the combinatory number of choosing M elements from N possibilities without replacement.

This method can obtain an exact explanation g if we evaluate all the possible examples of z, that is, 2^F evaluations of the black box f. As the number of evaluations required increases factorially with respect to the number of features, this nonstochastic approximation is unaffordable. That is why the general use of this method uses also a stochastic approximation generating a list of N different examples and solves the linear ridge regression as LIME does.

The generation of the neighborhood $N(x)$ is performed by sampling a value v from a random discrete variable X whose distribution is the following:

$$P[X = x] = \frac{1/(x+1)(M-x+1)}{\sum_{i=0}^{M} 1/(i+1)(M-i+1)}, \quad x = 0, \ldots, M, \tag{5}$$

that is, X is the random variable that assigns x the proportional probability of the weight that SHAP assigns to all the instances that exclude exactly x variables. The value v sampled is used to select randomly v features to exclude on this sample.

2.3. How to Define Features for LLE in Nontabular Data. For an explanation based on feature importance, it is very important to define what a feature is. In tabular data, a feature is defined naturally from the dataset itself. However, other types of data do not have this convenience, e.g., time series or images. In the case of time series, the minimum amount of information is obtained at each measurement timestep. In the case of images, we get it from each pixel. This has several associated problems:

(i) Generating exact explanations becomes an unaffordable task. In the case of SHAP, for a number of F features, 2^F evaluations of the black box are needed to generate the nonprobabilistic explanation. A generic ImageNet image has a size of $224 \cdot 224 = 50176$ pixels, resulting in 2^{50176} black-box evaluations in SHAP. Even in its probabilistic versions, a regression needs a large number of these evaluations to be reliable.

(ii) Explanations lose perspective. For a human being, a single pixel means nothing. In order to make a meaningful explanation, several pixels must be grouped together.

To solve these problems, some works use a division of the image into squares of the same size [12] while others use an unsupervised segmentation method to generate larger segment-size features [13].

2.4. How to Explain with LLE in Different Machine Learning Tasks. To explain an artificial intelligence model, it is necessary to take into account the task for which the model has been designed.

(i) In the regression task, each element of the output can be explained separately. Thanks to this, no output is dependent on any other and a separate analysis can be performed.

(ii) In the classification task, the output is usually a vector of probabilities with clear constraints that must be satisfied (each element must be greater than or equal to 0 and the sum of all of them must be 1). Furthermore, it is not just the class to which it is classified that has an influence, but also the degree of certainty with which it is classified into each class. Since the outputs are dependent on each other in this case, a joint analysis of the output must be carried out.

(iii) In the clustering task, an explanation can be carried out simply by some example or by some rule for each cluster [14]. Therefore, it is necessary to unify the concept of explanation within the clustering task.

Therefore, for each specific task, a different method of explanation must be developed. From now on, we focus on the task of classification, described formally in the following.

2.4.1. Classification Task Specifications. Let g be a local linear white-box-model where $g\colon F \longrightarrow Y_l$ over the logit space, $g(x) = Ax + B$. We define the signed importance matrix as the derivative matrix A^l, over the logit space. It should be noted that $A = A^l$.

To obtain the probability vector, we need to apply the softmax function, that is, $p = \mathrm{softmax}(Ax + B)$. We define $A^p = D(\mathrm{softmax}(g(x)))(x)$ where $D()$ is the derivative operator.

The component $a_{i,j}$ of matrix A^l and A^p will refer to the importance of feature i for class j over the logit and probability spaces, respectively.

Both matrices give us important and complementary information about the behavior of the white box g. The A^l matrix gives us absolute information about how the logits of all classes correspond to the original features. Additionally, the A^p matrix gives us information about the classes that are potentially most likely to be classified as, disregarding the least likely. This may provide us apparently contradictory information, as we show in the following example:

(i) Let $g\colon \mathcal{X} \longrightarrow \mathbb{R}^3$ be the white-box linear model of a multiclass problem of three classes on the logit regression and let x be the original example. Say $g(x) = (5, 3, -2)$ and, therefore, $\mathrm{softmax}(g(x)) = (95.17\%, 4.73\%, 0.08\%)$.

(ii) We now consider x', a neighbor of x with a perturbation on i feature, that produces $g(x') = (2.5, 1.5, -1)$ and, therefore, $\mathrm{softmax}(g(x')) = (71.52\%, 26.31\%, 2.15\%)$.

(iii) If we consider exclusively the logit approximation, it may be interpreted as feature i influences positively for classes 1 and 2 and negatively for class 3, with approximately the same intensity.

(iv) If we consider exclusively the probability approximation, feature i may have a positive influence for class 1, a negative influence for class 2, and, much less significantly, a negative influence for class 3.

From a global view point, each view point has its impact on the analysis. Thus, we define a new matrix $\mathscr{A}$ as the importance matrix and it is obtained as follows from the matrices A^l and A^p:

$$\mathscr{A}_{i,j} = \mathrm{sign}\left(A^l_{i,j}\right)\sqrt{\left|A^l_{i,j}\right| \cdot \left|A^p_{i,j}\right|}, \tag{6}$$

which attempts to combine the information of both matrices A^l and A^p. This matrix $\mathscr{A}$ has the sign of the logit matrix and the geometric mean of the intensity of importance of both matrices.

From the importance matrix $\mathscr{A}$, we define the relative importance matrix $\mathscr{A}_r$ as $\mathscr{A}/\max_{a_{i,j}\in\mathscr{A}}(|a_{i,j}|)$ and the normalized matrix that maintains 0 as 0 and transforms the value with the greater absolute value to 1 or −1, depending on the original sign of this specific value.

We define the absolute importance matrix $|\mathscr{A}|$ as the matrix of the terms $\mathscr{A}_r$ in absolute value, that is, $a_{i,j} = |a_{i,j}|$ for each coefficient i, j of matrix $\mathscr{A}$. Each term $a_{i,j}$ of $|A_{i,j}|$ is the absolute importance of feature i to the class j.

2.5. Proposed Frameworks to Compute LLE: Qualitative and Quantitative Approaches. All proposed explanations are based on different notions of what constitute an explanation and, therefore, are not directly comparable. In the literature, there are several proposals to compare explanations. In [15], another set of metrics is proposed to measure the quality of explanations. However, they are specialized in rules-based explanations. In [6], the LEAF framework is proposed, with also four different metrics to evaluate agnostically different explanation metrics, independent of the explanation generation method. It also offers a practical example of their use, evaluating the quality of different explanations. However, the theoretical development of this framework is not mathematically consistent, which leads to biased conclusions. On the one hand, qualitative measures are not objective, and on the other hand, poorly calibrated measures where the worst or best score is not achieved only by the worst or best option, respectively, result in saturations that equalize part of the examples, removing relevant information. In addition, it is desirable that these measures have good mathematical properties such as smoothness so that there are no abnormalities in practical cases that were not considered.

It is in this scenario where the need for a mathematically consistent and unbiased explanation evaluation framework arises. In addition, this framework must also provide a measure not only comparative but also giving an absolute idea of the good behavior of the explanation itself.

3. REVEL Framework

In this section, we propose a new explanation evaluation framework called REVEL framework, presenting five new metrics for assessing the quality of an explanation. In particular, for each metric proposed, we describe the qualitative aspect we want to measure with the metric and show how the formal definition measures this aspect. We also provide a guideline on how to interpret the metric. Finally, for each qualitative aspect, we make a theoretical comparison of each metric with other proposed metrics.

In Table 1, we summarize the metrics we propose and the qualitative aspect they measure.

3.1. Local Concordance. There are LLE methods guaranteeing the white-box explanation and the black-box model to match on the specific datapoint. However, these methods have a strong computational constraint, since they require a large number of evaluations of the black-box model. Other methods do not ensure the coincidence between white-box explanation and black-box model. Since the concordance between both is not guaranteed, it is possible that the class proposed is different from each other, which means the proposed explanations end up being inconsistent. We want to measure how much the explanation and the model are similar.

On the classification task of more than two classes, it is also necessary to consider jointly the whole probability vector. Our proposal also attempts to measure the smoothness from the min to the max concordance values, that is, only the min concordance should have a score of 0 and the max concordance should have a score of 1 on this metric.

We can easily abstract the loss function that evaluates our metric to consider vector distances among probability vectors:

$$\text{Local\_Concordance}\,(g) = 1 - \frac{|f(x) - g(x)|}{C}, \tag{7}$$

where $|\cdot|$ is a defined norm (1-norm, 2-norm, inf-norm...) and C is the maximum distance between two possible probability vectors. This term exists and is reached because the probability space is complete and the norm is continuous. Moreover, C is computed as $|u - v|$, where $u = (1, 0, \ldots, 0)$ and $v = (0, 1, 0, \ldots, 0)$, regardless of the norm.

This metric has the following qualities:

(i) Using C as the normalization factor makes our score well defined in the interval [0, 1], with the max concordance achieving 1 and the min concordance achieving 0, regardless of the number of classes in the dataset.

(ii) This metric considers the whole probability vector jointly and not just one coordinate of the probability vector.

3.1.1. Guideline. This metric measures how similar the explanation is to the black box in the original example. It is very important that this metric is close to 1. Otherwise, the proposed explanation does not explain what happens in the example itself.

3.1.2. Comparison. The analogous LEAF proposal local concordance is defined as $l(|f(x) - g(x)|)$, where $l(k) = \max(0, 1 - k)$ is the hinge loss function [16]. In contrast to our proposal, the use of the hinge function makes it non-smooth. It also does not assure that only the maximum discordance reaches the worst value of the metric. In conclusion, the LEAF proposal has inconsistencies that our proposal overcomes.

3.2. Local Fidelity. Local fidelity applies not to a classification task but a regression one. The main idea of this metric is how close is the white box g approximating the probabilities obtained by the black box f. We propose the mean concordance between probabilities of g and f obtained on the neighborhood $N(x)$, that is,

$$\text{Local\_Fidelity}\,(g) = \frac{1}{|N(x)|} \sum_{n \in N(x)} 1 - \frac{|f(n) - g(n)|}{C}. \tag{8}$$

This metric is an extension of the local concordance on x extended to its neighborhood $N(x)$. It is also well defined on the interval [0, 1], regardless of the number of classes in the dataset.

3.2.1. Guideline. This metric measures the similarity between the explanation and the black box in the neighborhood. This metric is essential to check that the tendency of the explanation is similar to the tendency of the black box. It must be close to 1 to obtain a good explanation.

3.2.2. Comparison. The analogous LEAF metric proposes to evaluate the resemblance between the white-box explanation and the black-box model in the proposed neighborhood $N(x)$ using the $F1$ metric.

(i) The LEAF proposal is a measure designed to evaluate classification problems. Since $N(x)$ is a neighborhood of x, most examples will, by continuity, be of the same class as x, resulting in an imbalance in $N(x)$.

(ii) This metric presents problems at decision borders. In a binary problem with threshold 0.5, let x' be an example of set $N(x)$ where $g(x') = 0.49$ and $f(x') = 0.51$. The $F1$ metric will penalize this example while actually the white box g mimics almost perfectly the undecidability of the black box f.

Table 1: Summary of the metrics developed by REVEL and the qualitative aspect they measure.

Name	What is evaluated
Local concordance	How similar is the LLE to the original black-box model on the original example
Local fidelity	How similar is the LLE to the original black-box model on a neighborhood of original example
Prescriptivity	How similar is the LLE to the original black-box model on the closest neighbor that changes the class of the original example
Conciseness	How brief and direct is the explanation
Robustness	How much two explanations generated by the same LLE generator differ

Our proposal has no problem with the imbalance dataset generated by $N(x)$ for the metric evaluation. Also, our metric is not biased by a threshold selection.

3.3. Prescriptivity. The main idea of prescriptivity is to test whether the white-box explanation g has correctly predicted the changes needed in the original example in order to change the original class.

Mathematically, let x be the original example, f be the black-box model, g be the white-box model mimicking f, and h be the changes needed on x to change the class predicted by the white box g. We propose the following prescriptivity metric:

$$\text{Prescriptivity}(g) = 1 - \frac{\|f(x+h) - g(x+h)\|}{C}, \tag{9}$$

where C is a normalization factor. This normalization factor is the same as in equation (7).

In our proposal, h is obtained by removing the presence of the most important positive features of the class predicted by the white box g on the example x. The algorithm ends when g assigns a different class to x and $x+h$, that is, $\text{argmax}(g(x)) \neq \text{argmax}(g(x+h))$.

This metric has the following properties:

(i) This prescriptivity proposal is defined as a vectorized proposal, so the metric has a global view of the whole output.

(ii) This metric obtains the maximum value 1 when both vectors u and v are equal and obtains the minimum value 0 when both vectors are in the maximum possible disagreement on this prescriptivity scenario. It is designed not to depend on the dimensions of the probability vectors either, so the metric is independent of the number of classes in the dataset.

(iii) This metric is not dependent of a boundary selection, nor it is dependent on a specified neighborhood $N(x)$.

3.3.1. Guideline. Prescriptivity challenges the explanation to propose an example far enough to change the prediction of the model but without losing predictive quality at this point. Indirectly, each explanation proposes an example x' different from the original example x whose prediction must be markedly different from that of x. Although the best possible score for this metric is 1, it is understandable that it does not reach the best score and serves more as a comparative metric between different explanation methods.

3.3.2. Comparison with LEAF. The prescriptivity metric is formally proposed in LEAF for a binary classification problem, where a fixed decision boundary is chosen. This decision boundary is the set $\mathscr{D}_g(y') = \{x \in \mathbb{R}_F : g(x) = y'\}$, that is, the set of points in the domain whose prediction by the white box is exactly y'.

On the LEAF proposal, the obtention of h is based on the closest projection of our example x on $\mathscr{D}_g(y')$. In reality, this is only possible if the features selected are real-valued. In case of binary data, this approximation cannot be achieved because each feature cannot process a real value. It is also dependent on a selection of a boundary y'.

LEAF proposes as prescriptivity metric the following function:

$$l\left(\frac{|f(x') - g(x')|}{C}\right), \tag{10}$$

where $l(\cdot)$ is the hinge loss function and $C = \max(y', 1-y')$ is a normalization factor, so that 1 means that x' lies at the boundary, and 0 means x' is at the furthest distance from the boundary. One may observe that by taking the absolute value, the measure both overshoots and undershoots the boundary as a loss of prescriptivity.

The LEAF proposal has different problems:

(i) This metric is designed for a single output variable. For classification problems, it is usual to obtain a vector of probabilities whose components are linked to each other and whose analysis must be done jointly.

(ii) Choosing a fixed y' value does not guarantee the change of class when we talk about nonbinary classification problems. In case of a classification problem of more than two classes, the majority class could have a 50% probability and other classes could share the rest of the probability equally. This results in a x' neighbor of x whose changes do not change the original class.

(iii) The proposed norm is restricted to the interval [0, 1] but not smoothly. Even if it is used a normalization parameter C, it is not clear if only the maximum possible disagreement results in a 0 score on this metric or if it is even reachable. It is reasonable for

this kind of metric to guarantee that the maximum disagreement obtains 0 as the worst score, and as agreement increases, the metric increases smoothly up to 1, the maximum score.

Our proposal does not show all of the different problems detected in the LEAF prescriptivity proposal, since our metric jointly measures the full probability vector, is not boundary dependent, and is well defined in the interval [0, 1], where it changes smoothly from worst case to the best one.

3.4. Conciseness. Conciseness measure aims to evaluate the brevity of the explanation. In our case, the less relevant features our explanation has, the more concise it should be.

We propose the following conciseness metric based on the absolute importance matrix $|\mathscr{A}|$, particularly in the vectors of importance of each feature. Let $v_i = (a_{i,1}, \ldots, a_{i,N})$ be the importance vector of feature i, where the coefficient $a_{i,j}$ is the i, j coefficient of matrix $|\mathscr{A}|$. We define the conciseness of the explanation proposed by the white box g as

$$\text{Conciseness}(g) = \frac{1}{f-1}\sum_{i=1}^{f} 1 - |v_i|_1, \tag{11}$$

which can be described as the mean irrelevance of the features. If we consider $|v_i|$ instead of $1 - |v_i|_1$, we would have the mean relevance of the features and the most concise method would have a score of $1/f - 1$. That is why we have reversed this term.

This metric has the following qualities:

(i) It rewards the use of few features with a high weight.

(ii) We have a general idea of how many features are important on the white box.

(iii) The best possible score is obtained if we have only one feature with absolute importance 1 and the rest with 0 absolute importance, in which case we would obtain 1 as conciseness. The worst case is obtained when we have all the features with 1 as absolute importance, in which case we would obtain 0 as conciseness. In addition, the number of classes has been taken into account, so that, regardless of the number of classes in the dataset, these max and min values are reachable.

(iv) We can compare explanations with different amount of features taken into account.

3.4.1. Guideline. This metric evaluates the ability of the explanation to focus on the most important features of an example and discard the less important ones. Depending on the complexity of the explanation we want, we may prefer greater or lesser conciseness. For instance, in image classification, the explanation to dismiss a large part of the image could be desired but not to have a single pixel explaining the complete decision of the model.

3.4.2. Comparison. LEAF proposes as conciseness a constraint for explanations, where it requires that explanations use exclusively k features. In the case of LIME, conciseness is a variable that we supply to the algorithm so that it restricts itself to choose a given number of features with nonzero importance. On the other hand, in the case of SHAP, the algorithm uses by default all available features and gives them an importance. In order to compare both methods, the LEAF framework proposes to select a default conciseness parameter k, the number of features to be used on the white-box explanation, and restrict both LIME and SHAP to use the top-k most important features.

As mentioned in the previous paragraph, the proposed conciseness is not a metric but a constraint on white-box explanation models. Moreover, the LEAF proposal does not leave the white-box models to decide whether a particular decision has been influenced by more or fewer features.

Our proposal, instead of a constraint, provides a metric to evaluate the conciseness of each white-box explanation.

3.5. Robustness over Explanations. A key point to consider is the variability of the methods used to generate explanations. It is desirable that independent explanations generated by the same method must be as similar as possible, since very different or even contradictory explanations would lead to mistrusting the method. In case of deterministic methods, this is ensured since there is just one proposed explanation. In case of nondeterministic methods, there are several proposed explanations, and therefore, we need to ensure that the explanations do not differ or even contradict each other.

To measure how two explanations g and g' differ, we propose two possible measures:

(i) First, we propose the cosine similarity between $\mathscr{A}_r$ and $\mathscr{A}'_r$, which are the relative importance matrices of g and g', respectively:

$$\text{sim}_{\cos}(g, g') = \frac{\mathscr{A}_r \cdot \mathscr{A}'_r}{|\mathscr{A}_r||\mathscr{A}'_r|}, \tag{12}$$

where $\cdot$ is the scalar product.

(ii) The metric proposed before based on the cosine similarity takes into account the direction of the matrices $\mathscr{A}$ and $\mathscr{A}'$ but not the magnitude. To take the magnitude also into account, we propose the following measure of similarity:

$$\text{similarity}'(g, g') = \left(\frac{\mathscr{A}_r \cdot \mathscr{A}'_r}{|\mathscr{A}_r||\mathscr{A}'_r|}\right) \cdot \left(1 - \left|\frac{\|\mathscr{A}_r\| - \|\mathscr{A}'_r\|}{\max(\|\mathscr{A}_r\|, \|\mathscr{A}'_r\|)}\right|\right), \tag{13}$$

that takes into account both direction and magnitude of the explanations. In case of the same magnitude, this similarity function is exactly the cosine similarity. In case of different magnitude, this similarity function has lesser punctuation than the

cosine similarity in case of a positive scalar product. In case of a negative scalar product, this score has also a lesser absolute value than the cosine similarity. In case of perpendicular explanation vectors, both metrics have a 0 score.

In both cases, as robustness, we propose the mathematical expectation of the chosen similarity of two different explanations g and g', that is:

$$\text{robustness}(G) = \mathbb{E}[\text{similarity}(g, g')], \tag{14}$$

where G is the set of all explanations that could be proposed by a certain explanation method such as LIME or SHAP. The expectation can be approximated by generating a given number of explanations and computing the mean of the similarities among explanations.

Those metrics have the following qualities:

(i) Both metrics take into account the weight of all features, so two explanations g and g' choosing a different most important feature would be punished by both metrics.

(ii) The second metric takes into account the magnitude of the importance matrix.

(iii) As both robustness metrics use bounded similarity functions, this metric consistently achieves the maximum and minimum possible in the best and worst case scenarios, respectively, regardless of the number of classes in the dataset.

3.5.1. Guideline. This metric does not evaluate a specific explanation but the method that generates them. All deterministic methods will score 1 in this metric since they always generate the same explanation. Therefore, this metric is designed to evaluate the robustness of nondeterministic methods. The closer this metric is to 1, the less the explanations generated by this method vary. It should be noted that this metric, due to the way it is designed, can give negative scores, which would indicate that the proposed explanations are contradictory.

3.5.2. Comparison. LEAF proposes the reiteration similarity metric, which measures how much two explanations generated by the same method vary by measuring the difference between the top-k features over several explanations.

(i) This metric depends directly on the conciseness constraint of the LEAF proposal.

(ii) This metric does not consider the importance of a feature, since it penalizes equally for choosing important and not so important features, not penalizing it.

(iii) This metric does not penalize choosing a "positive" important feature as "negative" and vice versa. Two different explanations can consider using the same feature i for their explanation but attributing positive importance to it in the first explanation and negative importance in the second, which is a clear contradiction. The similarity proposal does not see this example as a contradiction and does not penalize it.

Our proposed robustness metric does not depend on external constraints and does not have the shortcomings described above while still measuring the variation between generated explanations.

4. Experimental Setup

In this section, we describe the experimental setup we use in this work. The objective of this experimental section is to show how to implement the proposed measures, not to demonstrate the best performance of our measures compared to others. This demonstration must be done at the theoretical stage, as we do in Section 3. We first select four image datasets as benchmark where we train the models to explain. Finally, we fix some hyperparameters to compare different LLE aspects with the REVEL framework.

4.1. Benchmark Selection. The datasets selected as benchmarks are CIFAR10 [17], CIFAR100 [18], FashionMNIST [19], and EMNIST-balanced [20], which is a benchmark already used in [2] for explainability tasks. Table 2 shows a short description of each dataset.

4.2. General Training Pipeline. For this experiment, we chose the EfficientNet-B2 model [21] with the pretrained weights in the ImageNet dataset. Next, the network has been fine-tuned on the benchmark dataset for 100 epochs, 32 images per batch with the Adam optimizer [22] with learning rate $1e-5$, weight decay = 0.001, and AMSGrad = True. We randomly selected 10% of the training set as validation subset on which the loss is not computed. Over the 100-epoch models, we select the model whose performance on this validation subset is the best. As the objective of this work is the analysis of the metric behavior, we will not go deeper into the training of the network and we will set these parameters as default. In Table 3, we show the performance obtained by the model in the different test sets of the datasets used as benchmarks.

4.3. Local Linear Explanation Pipeline. In this section, with the purpose of generating a fair comparison, we fix as default some shared hyperparameters of the LLE generation models, explained below.

Number of neighbors (N): For each example of the test split, we will generate a different number of neighbor examples to explain the original example. In the experiments, N = 100, 200, 300, 400, 500, 600, 700, 800.

Neighbor generation ($N(x)$): we use a smart perturbation generator, where each neighbor is generated with a probability proportional to the weight associated to it in each method of explanation generation.

Table 2: Descriptive table of the benchmarks selected.

Dataset	Number of classes	Original image size	Training	Test	RGB
CIFAR10	10	$32 \cdot 32$	50.000	10.000	Yes
CIFAR100	100	$32 \cdot 32$	50.000	10.000	Yes
FashionMNIST	10	$28 \cdot 28$	60.000	10.000	No
EMNIST-balanced	47	$28 \cdot 28$	112.800	18.800	No

Table 3: Performance of the default model on the test sets of the selected datasets.

Dataset	Train/test partition	Classification model top-1 accuracy (test) (%)
CIFAR10	83.3%/16.7%	95.26
CIFAR100	83.3%/16.7%	81.84
FashionMNIST	86%/14%	94.25
EMNIST	86%/14%	90.66

Number of explanations generated (E): for each LLE method and each instance to be explained, we will generate 5 different explanations.

Number of features of each image ($F \cdot F$): we divide each image into square patches of size $224/F \cdot 224/F$, so each image will have $F \cdot F$ features.

Feature occlusion: to set a feature as occluded, we set the original patch from its original value to a neutral grey patch, that is, we set all pixel of the patch to 0.5 on each RGB channel.

4.4. On the Comparison between LEAF and REVEL. This paper presents REVEL as a proposal of theoretically robust measures for the evaluation of LLE explanations. The comparison with other measurement proposals, such as LEAF, should be carried out theoretically and not practically, since the measurements offered by the different proposals have nothing related to each other. That is why the comparison on this work is made exclusively on the theoretical proposal and not on the practical use cases.

5. Assessing Explanations Using REVEL: Use Cases

In this section, we propose three different scenarios in which REVEL can be used, thus demonstrating its analytical potential. These scenarios are as follows:

(i) Dependence of LIME on the number of features (Section 5.1): in this scenario, we study how much the number of patches into which we have divided the original image can influence, or if there is an ideal partition in which to divide the images.

(ii) Dependence of LIME on the number of black-box evaluations (Section 5.2): In this scenario, we analyze the number of black-box evaluations needed to generate a good-quality explanation. We also evaluate the trade-off between quality and time needed to generate a good explanation.

(iii) LIME vs. SHAP (Section 5.3): We compare the results obtained by the two state-of-the-art explanation generator models, LIME and SHAP, with the best configuration determined by the above scenarios. This scenario provides an idea about which explanation generator can offer us better explanations depending on their scores in each of the proposed metrics.

To better support our analysis on the experiments for each previous scenarios, we use the Shapiro test and Wilcoxon test from the SciPy stats library [23]. In particular, the Shapiro test has been used to check whether each experiment follows a normal distribution, and thus using a parametric test is appropriate (Tables 4–6). Since with an exception of a single experiment in Table 4 we can discard that the distributions are normal, we need the nonparametric Wilcoxon test to check that the distributions of different experiments have essentially different results. We considered a p value of 0.05 to discard the null hypothesis. For each use case, the Wilcoxon test tables used to perform the comparisons are cited.

5.1. Dependence of LIME on the Number of Features. In this section, we compare how LIME performs over different number of features. This comparison allows to perform both a general study and a study focusing on the image data type. At a general level, we analyze how the number of features influences the quality of the explanation. In the case of images, we use this study to determine the best performing granularity.

5.1.1. Local Concordance. In Figure 1 we note that, as a tendency, the local concordance score increases as more features are processed. As the number of features increases, the explanation method has more parameters to fit. Therefore, the model increases its performance on mimicking the black box on the original example. The difference remains significant (Tables 7–10).

Table 4: Shapiro test for the number of features.

Features	Metric	CIFAR10	CIFAR100	EMNIST	FashionMNIST
36	Local_Concordance	0.0000	0.0	0.0	0.0000
36	Local_Fidelity	0.0000	0.0	0.0	0.0000
36	Prescriptivity	0.0000	0.0	0.0	0.0000
36	Conciseness	0.0000	0.0	0.0	0.0000
36	Robustness	0.0000	0.0	0.0	0.0000
49	Local_Concordance	0.0000	0.0	0.0	0.0000
49	Local_Fidelity	0.0000	0.0	0.0	0.0000
49	Prescriptivity	0.0000	0.0	0.0	0.0000
49	Conciseness	0.0000	0.0	0.0	0.0000
49	Robustness	0.0000	0.0	0.0	0.0000
64	Local_Concordance	0.0000	0.0	0.0	0.0000
64	Local_Fidelity	0.0000	0.0	0.0	0.0000
64	Prescriptivity	0.0000	0.0	0.0	0.0000
64	Conciseness	0.0000	0.0	0.0	0.2323
64	Robustness	0.0000	0.0	0.0	0.0000
81	Local_Concordance	0.0000	0.0	0.0	0.0000
81	Local_Fidelity	0.0000	0.0	0.0	0.0000
81	Prescriptivity	0.0000	0.0	0.0	0.0000
81	Conciseness	0.0001	0.0	0.0	0.0000
81	Robustness	0.0000	0.0	0.0	0.0000
100	Local_Concordance	0.0000	0.0	0.0	0.0000
100	Local_Fidelity	0.0000	0.0	0.0	0.0000
100	Prescriptivity	0.0000	0.0	0.0	0.0000
100	Conciseness	0.0000	0.0	0.0	0.0000
100	Robustness	0.0000	0.0	0.0	0.0000
121	Local_Concordance	0.0000	0.0	0.0	0.0000
121	Local_Fidelity	0.0000	0.0	0.0	0.0000
121	Prescriptivity	0.0000	0.0	0.0	0.0000
121	Conciseness	0.0003	0.0	0.0	0.0001
121	Robustness	0.0000	0.0	0.0	0.0000
144	Local_Concordance	0.0000	0.0	0.0	0.0000
144	Local_Fidelity	0.0000	0.0	0.0	0.0000
144	Prescriptivity	0.0000	0.0	0.0	0.0000
144	Conciseness	0.0000	0.0	0.0	0.0000
144	Robustness	0.0000	0.0	0.0	0.0000

Table 5: Shapiro test for the number of max examples and the number of features.

Examples	Metric	CIFAR10	CIFAR100	EMNIST	FashionMNIST
100	Local_Concordance	0.0	0.0	0.0	0.0000
100	Local_Fidelity	0.0	0.0	0.0	0.0000
100	Prescriptivity	0.0	0.0	0.0	0.0000
100	Conciseness	0.0	0.0	0.0	0.0004
100	Robustness	0.0	0.0	0.0	0.0000
200	Local_Concordance	0.0	0.0	0.0	0.0000
200	Local_Fidelity	0.0	0.0	0.0	0.0000
200	Prescriptivity	0.0	0.0	0.0	0.0000
200	Conciseness	0.0	0.0	0.0	0.0001
200	Robustness	0.0	0.0	0.0	0.0000
300	Local_Concordance	0.0	0.0	0.0	0.0000
300	Local_Fidelity	0.0	0.0	0.0	0.0000
300	Prescriptivity	0.0	0.0	0.0	0.0000
300	Conciseness	0.0	0.0	0.0	0.0003
300	Robustness	0.0	0.0	0.0	0.0000
400	Local_Concordance	0.0	0.0	0.0	0.0000
400	Local_Fidelity	0.0	0.0	0.0	0.0000
400	Prescriptivity	0.0	0.0	0.0	0.0000
400	Conciseness	0.0	0.0	0.0	0.0000
400	Robustness	0.0	0.0	0.0	0.0000

Table 5: Continued.

Examples	Metric	CIFAR10	CIFAR100	EMNIST	FashionMNIST
500	Local_Concordance	0.0	0.0	0.0	0.0000
500	Local_Fidelity	0.0	0.0	0.0	0.0000
500	Prescriptivity	0.0	0.0	0.0	0.0000
500	Conciseness	0.0	0.0	0.0	0.0000
500	Robustness	0.0	0.0	0.0	0.0000
600	Local_Concordance	0.0	0.0	0.0	0.0000
600	Local_Fidelity	0.0	0.0	0.0	0.0000
600	Prescriptivity	0.0	0.0	0.0	0.0000
600	Conciseness	0.0	0.0	0.0	0.0000
600	Robustness	0.0	0.0	0.0	0.0000
700	Local_Concordance	0.0	0.0	0.0	0.0000
700	Local_Fidelity	0.0	0.0	0.0	0.0000
700	Prescriptivity	0.0	0.0	0.0	0.0000
700	Conciseness	0.0	0.0	0.0	0.0000
700	Robustness	0.0	0.0	0.0	0.0000
800	Local_Concordance	0.0	0.0	0.0	0.0000
800	Local_Fidelity	0.0	0.0	0.0	0.0000
800	Prescriptivity	0.0	0.0	0.0	0.0000
800	Conciseness	0.0	0.0	0.0	0.0000
800	Robustness	0.0	0.0	0.0	0.0000

Table 6: Shapiro test for the different XAI methods.

Method	Metric	CIFAR10	CIFAR100	EMNIST	FashionMNIST
LIME_2.0	Local_Concordance	0.0	0.0	0.0	0.0000
LIME_2.0	Local_Fidelity	0.0	0.0	0.0	0.0000
LIME_2.0	Prescriptivity	0.0	0.0	0.0	0.0000
LIME_2.0	Conciseness	0.0	0.0	0.0	0.0000
LIME_2.0	Robustness	0.0	0.0	0.0	0.0000
LIME_3.0	Local_Concordance	0.0	0.0	0.0	0.0000
LIME_3.0	Local_Fidelity	0.0	0.0	0.0	0.0000
LIME_3.0	Prescriptivity	0.0	0.0	0.0	0.0000
LIME_3.0	Conciseness	0.0	0.0	0.0	0.0000
LIME_3.0	Robustness	0.0	0.0	0.0	0.0000
LIME_4.0	Local_Concordance	0.0	0.0	0.0	0.0000
LIME_4.0	Local_Fidelity	0.0	0.0	0.0	0.0000
LIME_4.0	Prescriptivity	0.0	0.0	0.0	0.0000
LIME_4.0	Conciseness	0.0	0.0	0.0	0.0000
LIME_4.0	Robustness	0.0	0.0	0.0	0.0000
LIME_5.0	Local_Concordance	0.0	0.0	0.0	0.0000
LIME_5.0	Local_Fidelity	0.0	0.0	0.0	0.0000
LIME_5.0	Prescriptivity	0.0	0.0	0.0	0.0000
LIME_5.0	Conciseness	0.0	0.0	0.0	0.0000
LIME_5.0	Robustness	0.0	0.0	0.0	0.0000
LIME_6.0	Local_Concordance	0.0	0.0	0.0	0.0000
LIME_6.0	Local_Fidelity	0.0	0.0	0.0	0.0000
LIME_6.0	Prescriptivity	0.0	0.0	0.0	0.0000
LIME_6.0	Conciseness	0.0	0.0	0.0	0.0030
LIME_6.0	Robustness	0.0	0.0	0.0	0.0000
LIME_7.0	Local_Concordance	0.0	0.0	0.0	0.0000
LIME_7.0	Local_Fidelity	0.0	0.0	0.0	0.0000
LIME_7.0	Prescriptivity	0.0	0.0	0.0	0.0000
LIME_7.0	Conciseness	0.0	0.0	0.0	0.0050
LIME_7.0	Robustness	0.0	0.0	0.0	0.0000
LIME_8.0	Local_Concordance	0.0	0.0	0.0	0.0000
LIME_8.0	Local_Fidelity	0.0	0.0	0.0	0.0000
LIME_8.0	Prescriptivity	0.0	0.0	0.0	0.0000
LIME_8.0	Conciseness	0.0	0.0	0.0	0.0258
LIME_8.0	Robustness	0.0	0.0	0.0	0.0000

Table 6: Continued.

Method	Metric	CIFAR10	CIFAR100	EMNIST	FashionMNIST
Global SHAP	Local_Concordance	0.0	0.0	0.0	0.0000
Global SHAP	Local_Fidelity	0.0	0.0	0.0	0.0000
Global SHAP	Prescriptivity	0.0	0.0	0.0	0.0000
Global SHAP	Conciseness	0.0	0.0	0.0	0.0000
Global SHAP	Robustness	0.0	0.0	0.0	0.0000
Local SHAP	Local_Concordance	0.0	0.0	0.0	0.0000
Local SHAP	Local_Fidelity	0.0	0.0	0.0	0.0000
Local SHAP	Prescriptivity	0.0	0.0	0.0	0.0000
Local SHAP	Conciseness	0.0	0.0	0.0	0.0000
Local SHAP	Robustness	0.0	0.0	0.0	0.0000

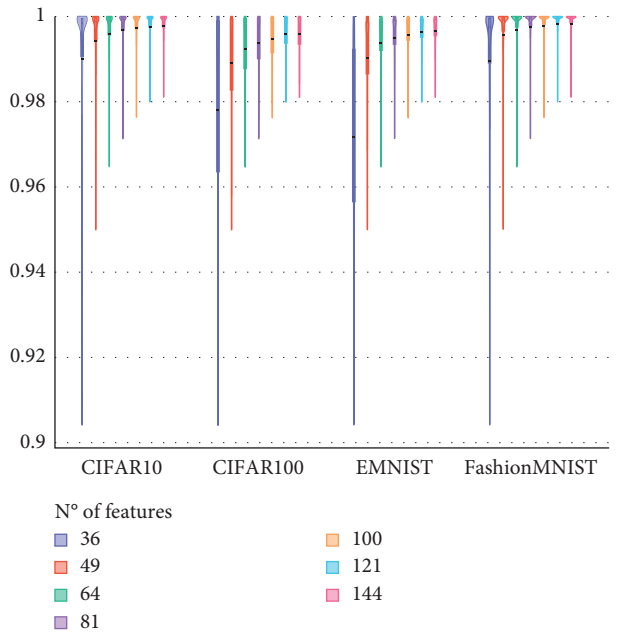

Figure 1: Performance of LIME methods grouped over number of features used in the local concordance metric.

Table 7: Wilcoxon test for the number of features and the local concordance metric in the CIFAR10 dataset.

	36	49	64	81	100	121	144
36	—	—	—	—	—	—	—
49	0.000205	—	—	—	—	—	—
64	0.0	0.002105	—	—	—	—	—
81	0.0	0.0	0.0	—	—	—	—
100	0.0	0.0	0.0	0.00482	—	—	—
121	0.0	0.0	0.0	0.000005	0.050236	—	—
144	0.0	0.0	0.0	0.0	0.087729	0.293829	—

Table 8: Wilcoxon test for the number of features and the local concordance metric in the CIFAR100 dataset.

	36	49	64	81	100	121	144
36	—	—	—	—	—	—	—
49	0.121167	—	—	—	—	—	—
64	0.000003	0.001704	—	—	—	—	—
81	0.0	0.0	0.0	—	—	—	—
100	0.0	0.0	0.0	0.0	—	—	—
121	0.0	0.0	0.0	0.0	0.0	—	—
144	0.0	0.0	0.0	0.0	0.0	0.157955	—

Table 9: Wilcoxon test for the number of features and the local concordance metric in the FashionMNIST dataset.

	36	49	64	81	100	121	144
36	—	—	—	—	—	—	—
49	0.0	—	—	—	—	—	—
64	0.0	0.0	—	—	—	—	—
81	0.0	0.0	0.000177	—	—	—	—
100	0.0	0.0	0.0	0.004466	—	—	—
121	0.0	0.0	0.0	0.000022	0.173217	—	—
144	0.0	0.0	0.0	0.0	0.009454	0.183706	—

Table 10: Wilcoxon test for the number of features and the local concordance metric in the EMNIST dataset.

	36	49	64	81	100	121	144
36	—	—	—	—	—	—	—
49	0.0	—	—	—	—	—	—
64	0.0	0.0	—	—	—	—	—
81	0.0	0.0	0.0	—	—	—	—
100	0.0	0.0	0.0	0.0	—	—	—
121	0.0	0.0	0.0	0.0	0.0	—	—
144	0.0	0.0	0.0	0.0	0.0	0.877998	—

5.1.2. Local Fidelity. In Figure 2, we note a tendency similar to the local concordance. That is, local fidelity increases the more features we use. This is natural since the neighbors where we are evaluating local fidelity are closer to the original example the more features we use. The differences remain significant except for 121 and 144 features (Tables 11–14).

5.1.3. Prescriptivity. In Figure 3, in contrast to the local concordance and local fidelity metrics, a different pattern arises, where as the number of features increases, the prescriptivity metric gets worse. Prescriptivity not only evaluates how well the explanation mimics the black box in areas near the original example but also evaluates the proposed changes to the white box. The fewer the features considered in the explanation, the fewer the changes necessary to change the predicted class. Thus, the explanation has less problems in finding the necessary features for the class to change. On the other hand, we must pay attention to where the differences in this metric are significant. In CIFAR10, FashionMNIST, and EMNIST, the results are significantly different while in CIFAR100, there are cases where they are not (Tables 15–18).

5.1.4. Conciseness. In Figure 4, we note a tendency to increase conciseness as the granularity increases. However, we observe that before this increase, conciseness decreases with 64 features. This seems to indicate that the higher the number of features, the better the performance. However, it can also be interpreted as an overfitting of the explanation and that the minimum amount of information that can be obtained from the image is by separating it into 64 different features and that a higher granularity overfits the model. It should be added that in almost all cases, there are significant differences (Tables 19–22). Even so, a study with images of various resolutions should be done because it could depend on the information contained on each patch.

5.1.5. Robustness. In Figure 5, we observe that the more features the models use, the more unstable the method becomes. Having more features to evaluate leads to more uncertainty in the choice of explanations. All the experiments present significant differences (Tables 23–26).

5.1.6. Global Conclusion. We appreciate that the higher the number of features, the better the local performance. This is an expected result since it is biased by the neighborhood we have chosen to calculate the local fidelity. Therefore, we should focus on the rest of the metrics. In the prescriptivity calculation, we see that the more the features are, the worse the result is obtained. In contrast, the more features we see, the more concise the methods are, discarding more unimportant features. Finally, we appreciate that LIME loses robustness the more features we use. This is due to the fact that the more features we use, the more likely it is that the explanation will use a larger set of features.

5.2. Dependence of LIME on the Number of Black-Box Evaluations. In this section, we will evaluate how important the number of black-box evaluations is over the LIME methods. This study is critical since black-box evaluations are considered the biggest bottleneck of black-box explainability methods. Although it is desirable to be able to evaluate the black-box function as many times as possible, there must be a trade-off between the quality of the explanation and the time it takes to generate it.

5.2.1. Local Concordance. In Figure 6, we can appreciate that increasing the number of black-box evaluations does not change the local concordance score significantly. Also, if we look at absolute values, we realize that we obtain significantly high values. This is due to the fact that the sampling used by LIME is very stable in picking the neighbors close to the original example. The fact that most experiments show no

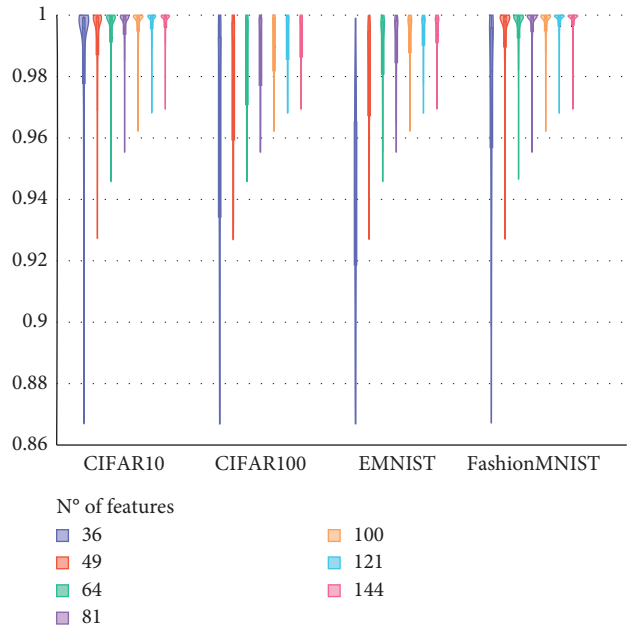

Figure 2: Performance of LIME methods grouped over number of features used in the local fidelity metric.

Table 11: Wilcoxon test for the number of features and the local fidelity metric in the CIFAR10 dataset.

	36	49	64	81	100	121	144
36	—	—	—	—	—	—	—
49	0.0	—	—	—	—	—	—
64	0.0	0.000288	—	—	—	—	—
81	0.0	0.0	0.0	—	—	—	—
100	0.0	0.0	0.0	0.005839	—	—	—
121	0.0	0.0	0.0	0.0	0.011812	—	—
144	0.0	0.0	0.0	0.0	0.00005	0.022145	—

Table 12: Wilcoxon test for the number of features and the local fidelity metric in the CIFAR100 dataset.

	36	49	64	81	100	121	144
36	—	—	—	—	—	—	—
49	0.0	—	—	—	—	—	—
64	0.0	0.0	—	—	—	—	—
81	0.0	0.0	0.0	—	—	—	—
100	0.0	0.0	0.0	0.0	—	—	—
121	0.0	0.0	0.0	0.0	0.0	—	—
144	0.0	0.0	0.0	0.0	0.0	0.386582	—

Table 13: Wilcoxon test for the number of features and the local fidelity metric in the FashionMNIST dataset.

	36	49	64	81	100	121	144
36	—	—	—	—	—	—	—
49	0.0	—	—	—	—	—	—
64	0.0	0.0	—	—	—	—	—
81	0.0	0.0	0.0	—	—	—	—
100	0.0	0.0	0.0	0.291572	—	—	—
121	0.0	0.0	0.0	0.0	0.000028	—	—
144	0.0	0.0	0.0	0.0	0.0	0.376673	—

Table 14: Wilcoxon test for the number of features and the local fidelity metric in the EMNIST dataset.

	36	49	64	81	100	121	144
36	—	—	—	—	—	—	—
49	0.0	—	—	—	—	—	—
64	0.0	0.0	—	—	—	—	—
81	0.0	0.0	0.0	—	—	—	—
100	0.0	0.0	0.0	0.0	—	—	—
121	0.0	0.0	0.0	0.0	0.000004	—	—
144	0.0	0.0	0.0	0.0	0.0	0.241494	—

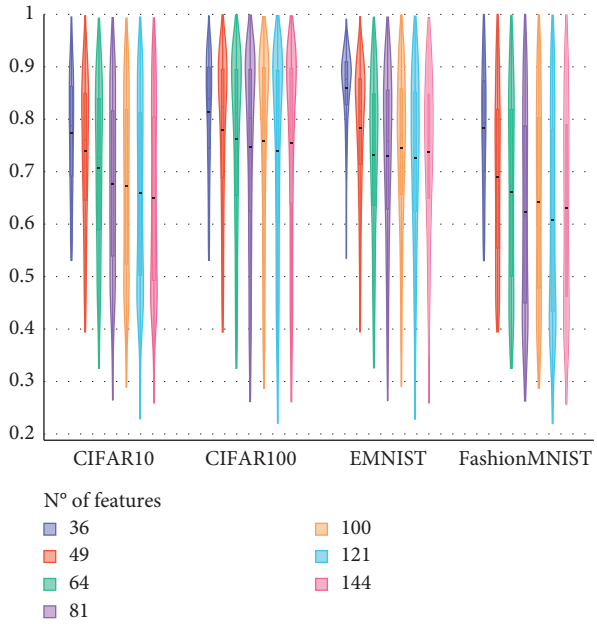

Figure 3: Performance of LIME methods grouped over number of features used in the prescriptivity metric.

Table 15: Wilcoxon test for the number of features and the prescriptivity metric in the CIFAR10 dataset.

	36	49	64	81	100	121	144
36	—	—	—	—	—	—	—
49	0.000023	—	—	—	—	—	—
64	0.0	0.0	—	—	—	—	—
81	0.0	0.0	0.0	—	—	—	—
100	0.0	0.0	0.0	0.002282	—	—	—
121	0.0	0.0	0.0	0.0	0.0	—	—
144	0.0	0.0	0.0	0.0	0.0	0.001164	—

Table 16: Wilcoxon test for the number of features and the prescriptivity metric in the CIFAR100 dataset.

	36	49	64	81	100	121	144
36	—	—	—	—	—	—	—
49	0.0	—	—	—	—	—	—
64	0.0	0.017497	—	—	—	—	—
81	0.0	0.0	0.000215	—	—	—	—
100	0.0	0.0	0.000713	0.705141	—	—	—
121	0.0	0.0	0.0	0.0	0.0	—	—
144	0.0	0.0	0.0	0.018171	0.002832	0.00614	—

Table 17: Wilcoxon test for the number of features and the prescriptivity metric in the FashionMNIST dataset.

	36	49	64	81	100	121	144
36	—	—	—	—	—	—	—
49	0.0	—	—	—	—	—	—
64	0.0	0.000801	—	—	—	—	—
81	0.0	0.0	0.0	—	—	—	—
100	0.0	0.0	0.0	0.021647	—	—	—
121	0.0	0.0	0.0	0.0	0.0	—	—
144	0.0	0.0	0.0	0.031498	0.000021	0.000167	—

Table 18: Wilcoxon test for the number of features and the prescriptivity metric in the EMNIST dataset.

	36	49	64	81	100	121	144
36	—	—	—	—	—	—	—
49	0.0	—	—	—	—	—	—
64	0.0	0.0	—	—	—	—	—
81	0.0	0.0	0.121733	—	—	—	—
100	0.0	0.0	0.192885	0.056783	—	—	—
121	0.0	0.0	0.0	0.0	0.0	—	—
144	0.0	0.0	0.000001	0.000004	0.0	0.820603	—

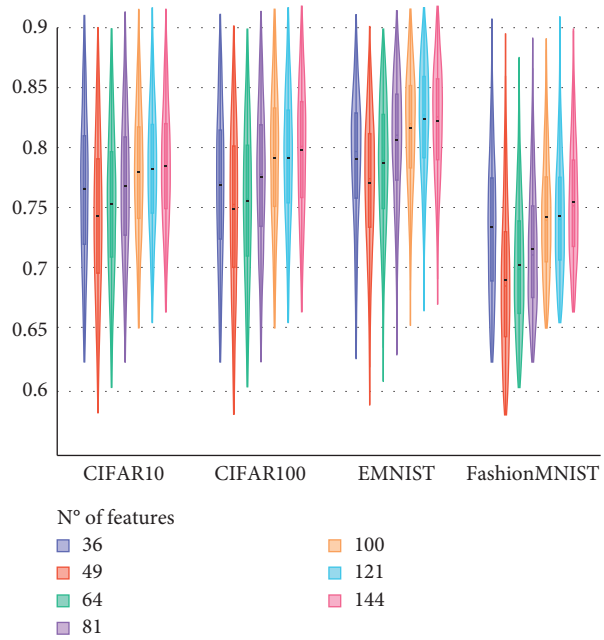

Figure 4: Performance of LIME methods grouped over number of features used in the conciseness metric.

Table 19: Wilcoxon test for the number of features and the conciseness metric in the CIFAR10 dataset.

	36	49	64	81	100	121	144
36	—	—	—	—	—	—	—
49	0.0	—	—	—	—	—	—
64	0.0	0.0	—	—	—	—	—
81	0.000006	0.0	0.0	—	—	—	—
100	0.0	0.0	0.0	0.0	—	—	—
121	0.0	0.0	0.0	0.0	0.000001	—	—
144	0.0	0.0	0.0	0.0	0.0	0.000006	—

Table 20: Wilcoxon test for the number of features and the conciseness metric in the CIFAR100 dataset.

	36	49	64	81	100	121	144
36	—	—	—	—	—	—	—
49	0.0	—	—	—	—	—	—
64	0.0	0.0	—	—	—	—	—
81	0.0	0.0	0.0	—	—	—	—
100	0.0	0.0	0.0	0.0	—	—	—
121	0.0	0.0	0.0	0.0	0.488279	—	—
144	0.0	0.0	0.0	0.0	0.0	0.0	—

Table 21: Wilcoxon test for the number of features and the conciseness metric in the FashionMNIST dataset.

	36	49	64	81	100	121	144
36	—	—	—	—	—	—	—
49	0.0	—	—	—	—	—	—
64	0.0	0.0	—	—	—	—	—
81	0.0	0.0	0.0	—	—	—	—
100	0.0	0.0	0.0	0.0	—	—	—
121	0.0	0.0	0.0	0.0	0.582463	—	—
144	0.0	0.0	0.0	0.0	0.0	0.0	—

Table 22: Wilcoxon test for the number of features and the conciseness metric in the EMNIST dataset.

	36	49	64	81	100	121	144
36	—	—	—	—	—	—	—
49	0.0	—	—	—	—	—	—
64	0.0	0.0	—	—	—	—	—
81	0.0	0.0	0.0	—	—	—	—
100	0.0	0.0	0.0	0.0	—	—	—
121	0.0	0.0	0.0	0.0	0.0	—	—
144	0.0	0.0	0.0	0.0	0.0	0.708112	—

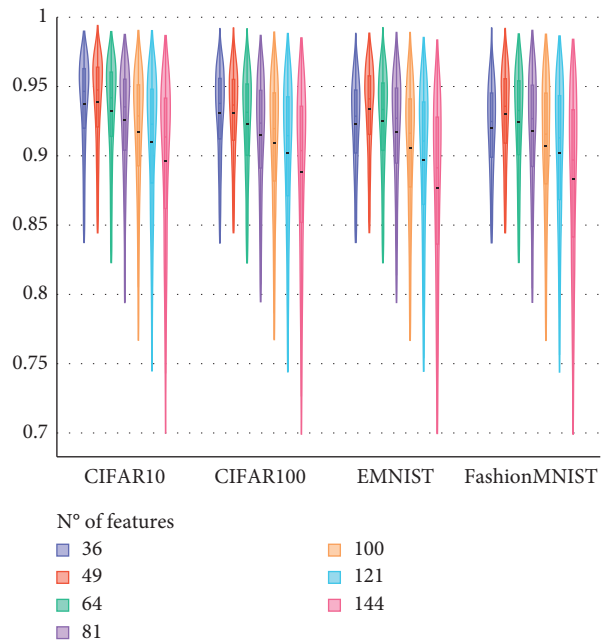

Figure 5: Performance of LIME methods grouped over number of features used in the robustness metric.

Table 23: Wilcoxon test for the number of features and the robustness metric in the CIFAR10 dataset.

	36	49	64	81	100	121	144
36	—	—	—	—	—	—	—
49	0.385861	—	—	—	—	—	—
64	0.0	0.0	—	—	—	—	—
81	0.0	0.0	0.0	—	—	—	—
100	0.0	0.0	0.0	0.0	—	—	—
121	0.0	0.0	0.0	0.0	0.0	—	—
144	0.0	0.0	0.0	0.0	0.0	0.0	—

Table 24: Wilcoxon test for the number of features and the robustness metric in the CIFAR100 dataset.

	36	49	64	81	100	121	144
36	—	—	—	—	—	—	—
49	0.0	—	—	—	—	—	—
64	0.0	0.0	—	—	—	—	—
81	0.0	0.0	0.0	—	—	—	—
100	0.0	0.0	0.0	0.0	—	—	—
121	0.0	0.0	0.0	0.0	0.0	—	—
144	0.0	0.0	0.0	0.0	0.0	0.0	—

Table 25: Wilcoxon test for the number of features and the robustness metric in the FashionMNIST dataset.

	36	49	64	81	100	121	144
36	—	—	—	—	—	—	—
49	0.0	—	—	—	—	—	—
64	0.0	0.0	—	—	—	—	—
81	0.0	0.0	0.0	—	—	—	—
100	0.0	0.0	0.0	0.0	—	—	—
121	0.0	0.0	0.0	0.0	0.0	—	—
144	0.0	0.0	0.0	0.0	0.0	0.0	—

Table 26: Wilcoxon test for the number of features and the robustness metric in the EMNIST dataset.

	36	49	64	81	100	121	144
36	—	—	—	—	—	—	—
49	0.0	—	—	—	—	—	—
64	0.0	0.0	—	—	—	—	—
81	0.0	0.0	0.0	—	—	—	—
100	0.0	0.0	0.0	0.0	—	—	—
121	0.0	0.0	0.0	0.0	0.0	—	—
144	0.0	0.0	0.0	0.0	0.0	0.0	—

significant differences between them corroborates this statement (Tables 27–30).

5.2.2. Local Fidelity. In Figure 7, we appreciate that, in this case, the more the evaluations of the black box, the better the result. We may expect that by randomly generating more neighbors, we obtain a better score in the neighborhood of the original example. However, as in local concordance metric, the differences in the experiments are not significant, corroborating the hypothesis that LIME is very stable near the example to be explained (Tables 31–34).

5.2.3. Prescriptivity. In Figure 8, we observe that the number of evaluations is not a differentiating factor. LIME proposes a series of changes that consistently change the prediction of the model by the same amount approximately. This hypothesis is corroborated by the fact that the experiments show no significant differences between them (Tables 35–38).

5.2.4. Conciseness. In Figure 9, we observe that the conciseness metric is influenced by the number of evaluations of the black box, making it less variable. Thus, LIME methods propose on average the same percentage of important features although increasing the number of evaluations tends to obtain less variable results, which is the main goal of increasing the number of maximum evaluation of black-box evaluations. Differences between experiments end up being significant when there is a difference in the number of

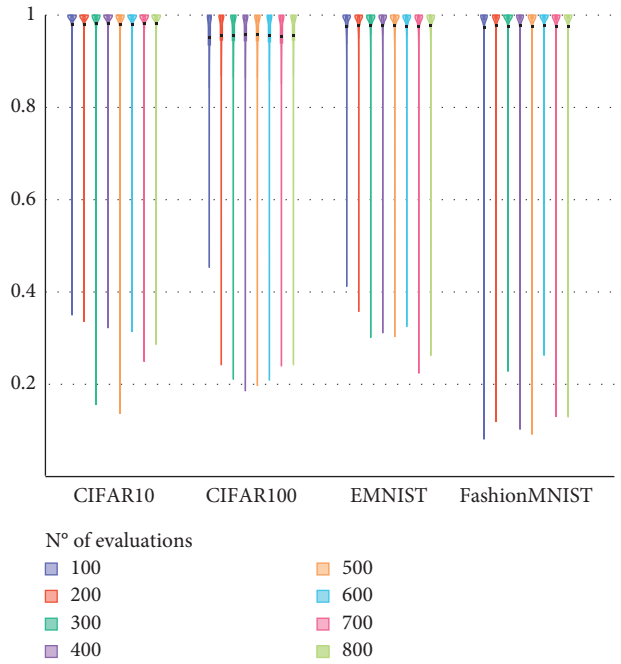

Figure 6: Performance of LIME methods grouped over max number of evaluations in the local concordance metric.

Table 27: Wilcoxon test for the number of max examples and the local concordance metric in the CIFAR10 dataset.

	100	200	300	400	500	600	700	800
100	—	—	—	—	—	—	—	—
200	0.006679	—	—	—	—	—	—	—
300	0.000004	0.079712	—	—	—	—	—	—
400	0.000005	0.110028	0.615945	—	—	—	—	—
500	0.000018	0.057421	0.967871	0.970798	—	—	—	—
600	0.000207	0.410788	0.292255	0.343868	0.775002	—	—	—
700	0.000008	0.095581	0.708867	0.494841	0.684628	0.580048	—	—
800	0.000088	0.202669	0.328229	0.394013	0.942875	0.997817	0.645379	—

Table 28: Wilcoxon test for the number of max examples and the local concordance metric in the CIFAR100 dataset.

	100	200	300	400	500	600	700	800
100	—	—	—	—	—	—	—	—
200	0.001844	—	—	—	—	—	—	—
300	0.000008	0.805911	—	—	—	—	—	—
400	0.0	0.051266	0.351842	—	—	—	—	—
500	0.000122	0.559817	0.769675	0.707414	—	—	—	—
600	0.00078	0.971505	0.50302	0.098521	0.632858	—	—	—
700	0.000566	0.652594	0.570233	0.208287	0.40554	0.70078	—	—
800	0.000011	0.136901	0.53912	0.639846	0.70536	0.250579	0.057151	—

Table 29: Wilcoxon test for the number of max examples and the local concordance metric in the FashionMNIST dataset.

	100	200	300	400	500	600	700	800
100	—	—	—	—	—	—	—	—
200	0.003203	—	—	—	—	—	—	—
300	0.000084	0.507544	—	—	—	—	—	—
400	0.000251	0.46633	0.442869	—	—	—	—	—
500	0.000004	0.172619	0.766257	0.790495	—	—	—	—
600	0.000021	0.452418	0.180001	0.849488	0.955108	—	—	—
700	0.000468	0.461356	0.972145	0.581572	0.67944	0.573234	—	—
800	0.0	0.091889	0.299545	0.762933	0.741445	0.376171	0.393368	—

Table 30: Wilcoxon test for the number of max examples and the local concordance metric in the EMNIST dataset.

	100	200	300	400	500	600	700	800
100	—	—	—	—	—	—	—	—
200	0.0	—	—	—	—	—	—	—
300	0.0	0.17739	—	—	—	—	—	—
400	0.0	0.915154	0.146136	—	—	—	—	—
500	0.0	0.010281	0.379468	0.135889	—	—	—	—
600	0.0	0.374906	0.631138	0.863129	0.187004	—	—	—
700	0.0	0.913341	0.229467	0.939601	0.078328	0.706811	—	—
800	0.0	0.09263	0.761107	0.749104	0.347307	0.941701	0.879176	—

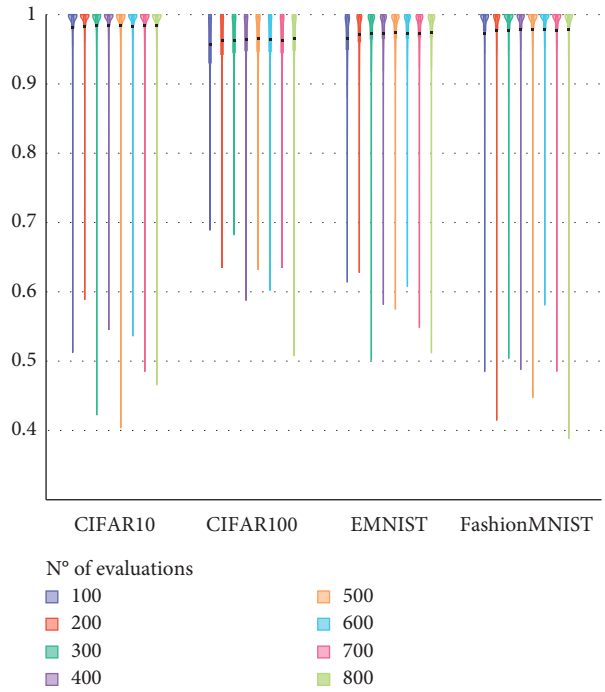

Figure 7: Performance of LIME methods grouped over max number of evaluations in the local fidelity metric.

Table 31: Wilcoxon test for the number of max examples and the local fidelity metric in the CIFAR10 dataset.

	100	200	300	400	500	600	700	800
100	—	—	—	—	—	—	—	—
200	0.00383	—	—	—	—	—	—	—
300	0.000063	0.029227	—	—	—	—	—	—
400	0.0	0.044276	0.948819	—	—	—	—	—
500	0.000003	0.00873	0.484636	0.762416	—	—	—	—
600	0.00003	0.142469	0.66191	0.453485	0.504563	—	—	—
700	0.0	0.006484	0.389521	0.977929	0.583729	0.375596	—	—
800	0.000027	0.23166	0.807472	0.398913	0.686507	0.925547	0.247536	—

Table 32: Wilcoxon test for the number of max examples and the local fidelity metric in the CIFAR100 dataset.

	100	200	300	400	500	600	700	800
100	—	—	—	—	—	—	—	—
200	0.0	—	—	—	—	—	—	—
300	0.0	0.116586	—	—	—	—	—	—
400	0.0	0.000094	0.033743	—	—	—	—	—
500	0.0	0.004109	0.13314	0.674681	—	—	—	—
600	0.0	0.003937	0.110797	0.621707	0.820858	—	—	—
700	0.0	0.021621	0.372432	0.740649	0.979647	0.40044	—	—
800	0.0	0.000007	0.002837	0.334775	0.087606	0.124636	0.01012	—

Table 33: Wilcoxon test for the number of max examples and the local fidelity metric in the FashionMNIST dataset.

	100	200	300	400	500	600	700	800
100	—	—	—	—	—	—	—	—
200	0.021311	—	—	—	—	—	—	—
300	0.000438	0.333243	—	—	—	—	—	—
400	0.001385	0.54142	0.31622	—	—	—	—	—
500	0.0	0.039717	0.246309	0.467611	—	—	—	—
600	0.000009	0.172931	0.100948	0.673224	0.597415	—	—	—
700	0.000141	0.28358	0.731288	0.875086	0.956504	0.559957	—	—
800	0.0	0.026779	0.071696	0.215298	0.434691	0.305927	0.282759	—

Table 34: Wilcoxon test for the number of max examples and the local fidelity metric in the EMNIST dataset.

	100	200	300	400	500	600	700	800
100	—	—	—	—	—	—	—	—
200	0.0	—	—	—	—	—	—	—
300	0.0	0.008746	—	—	—	—	—	—
400	0.0	0.005612	0.841771	—	—	—	—	—
500	0.0	0.000004	0.02384	0.224304	—	—	—	—
600	0.0	0.000002	0.026052	0.168841	0.823687	—	—	—
700	0.0	0.000001	0.023204	0.05079	0.969599	0.937187	—	—
800	0.0	0.000001	0.020001	0.095514	0.821418	0.689143	0.489925	—

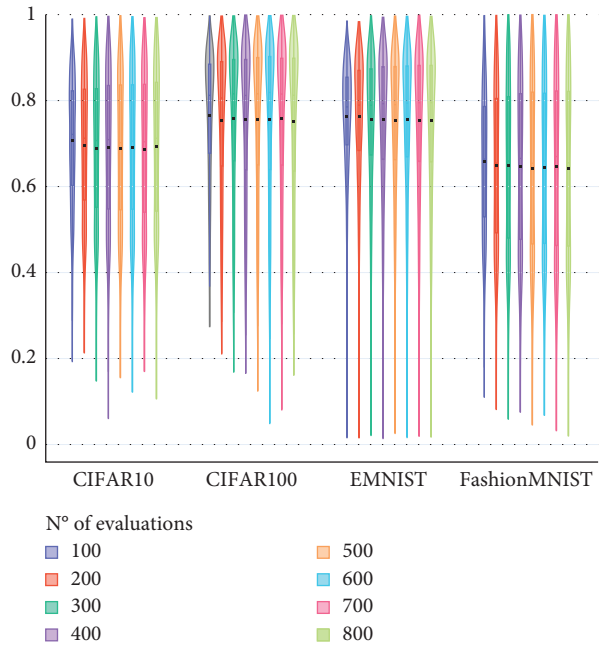

Figure 8: Performance of LIME methods grouped over max number of evaluations in the prescriptivity metric.

Table 35: Wilcoxon test for the number of max examples and the prescriptivity metric in the CIFAR10 dataset.

	100	200	300	400	500	600	700	800
100	—	—	—	—	—	—	—	—
200	0.5842	—	—	—	—	—	—	—
300	0.907305	0.201728	—	—	—	—	—	—
400	0.014904	0.000051	0.000592	—	—	—	—	—
500	0.000688	0.0	0.000001	0.072502	—	—	—	—
600	0.0	0.0	0.0	0.000012	0.004546	—	—	—
700	0.0	0.0	0.0	0.000001	0.000786	0.18838	—	—
800	0.0	0.0	0.0	0.0	0.0	0.0	0.000248	—

Table 36: Wilcoxon test for the number of max examples and the prescriptivity metric in the CIFAR100 dataset.

	100	200	300	400	500	600	700	800
100	—	—	—	—	—	—	—	—
200	0.059451	—	—	—	—	—	—	—
300	0.0	0.000231	—	—	—	—	—	—
400	0.0	0.000001	0.179442	—	—	—	—	—
500	0.0	0.0	0.000015	0.007755	—	—	—	—
600	0.0	0.0	0.000024	0.02083	0.539939	—	—	—
700	0.0	0.0	0.0	0.000002	0.049246	0.031741	—	—
800	0.0	0.0	0.000009	0.009774	0.794275	0.970759	0.041494	—

Table 37: Wilcoxon test for the number of max examples and the prescriptivity metric in the FashionMNIST dataset.

	100	200	300	400	500	600	700	800
100	—	—	—	—	—	—	—	—
200	0.534507	—	—	—	—	—	—	—
300	0.112968	0.020653	—	—	—	—	—	—
400	0.001432	0.000023	0.052142	—	—	—	—	—
500	0.001171	0.000052	0.010819	0.625029	—	—	—	—
600	0.000001	0.0	0.00002	0.010288	0.091626	—	—	—
700	0.0	0.0	0.0	0.000534	0.001308	0.220096	—	—
800	0.0	0.0	0.000012	0.005565	0.011733	0.613277	0.530278	—

Table 38: Wilcoxon test for the number of max examples and the prescriptivity metric in the EMNIST dataset.

	100	200	300	400	500	600	700	800
100	—	—	—	—	—	—	—	—
200	0.0	—	—	—	—	—	—	—
300	0.0	0.000001	—	—	—	—	—	—
400	0.0	0.0	0.000032	—	—	—	—	—
500	0.0	0.0	0.0	0.010922	—	—	—	—
600	0.0	0.0	0.0	0.0	0.001549	—	—	—
700	0.0	0.0	0.0	0.000202	0.059805	0.619722	—	—
800	0.0	0.0	0.0	0.000004	0.014507	0.939177	0.423023	—

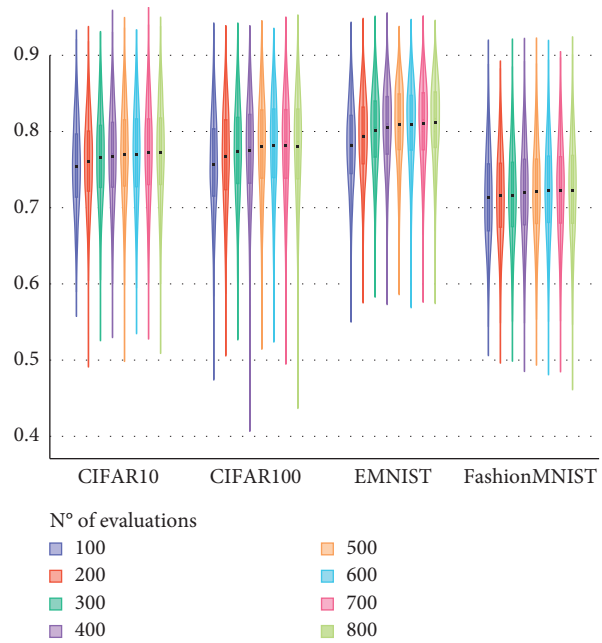

Figure 9: Performance of LIME methods grouped over max number of evaluations in the conciseness metric.

Table 39: Wilcoxon test for the number of max examples and the conciseness metric in the CIFAR10 dataset.

	100	200	300	400	500	600	700	800
100	—	—	—	—	—	—	—	—
200	0.0	—	—	—	—	—	—	—
300	0.0	0.609983	—	—	—	—	—	—
400	0.0	0.102388	0.185725	—	—	—	—	—
500	0.000003	0.598608	0.735026	0.165057	—	—	—	—
600	0.0	0.258252	0.286917	0.853349	0.325011	—	—	—
700	0.000016	0.588461	0.685646	0.251913	0.714545	0.352133	—	—
800	0.000004	0.432255	0.524231	0.490735	0.478845	0.650608	0.603819	—

Table 40: Wilcoxon test for the number of max examples and the conciseness metric in the CIFAR100 dataset.

	100	200	300	400	500	600	700	800
100	—	—	—	—	—	—	—	—
200	0.000402	—	—	—	—	—	—	—
300	0.024399	0.294329	—	—	—	—	—	—
400	0.002669	0.243941	0.055526	—	—	—	—	—
500	0.733905	0.058091	0.549324	0.008971	—	—	—	—
600	0.269944	0.361222	0.632872	0.077468	0.350272	—	—	—
700	0.242506	0.795124	0.677092	0.312538	0.205854	0.708336	—	—
800	0.053038	0.68238	0.44231	0.291114	0.17388	0.380468	0.937535	—

Table 41: Wilcoxon test for the number of max examples and the conciseness metric in the FashionMNIST dataset.

	100	200	300	400	500	600	700	800
100	—	—	—	—	—	—	—	—
200	0.0	—	—	—	—	—	—	—
300	0.0	0.000004	—	—	—	—	—	—
400	0.0	0.0	0.47811	—	—	—	—	—
500	0.0	0.0	0.144291	0.687725	—	—	—	—
600	0.0	0.000003	0.585043	0.613311	0.416136	—	—	—
700	0.0	0.0	0.156878	0.621658	0.648151	0.121491	—	—
800	0.0	0.0	0.037749	0.264671	0.376721	0.051915	0.631084	—

Table 42: Wilcoxon test for the number of max examples and the conciseness metric in the EMNIST dataset.

	100	200	300	400	500	600	700	800
100	—	—	—	—	—	—	—	—
200	0.336169	—	—	—	—	—	—	—
300	0.000386	0.007551	—	—	—	—	—	—
400	0.000002	0.000507	0.40222	—	—	—	—	—
500	0.0	0.000007	0.053632	0.29106	—	—	—	—
600	0.0	0.000015	0.104366	0.404215	0.64368	—	—	—
700	0.0	0.0	0.001188	0.021228	0.293383	0.166263	—	—
800	0.0	0.0	0.000002	0.000061	0.002865	0.00126	0.276616	—

evaluations between one experiment and another (Tables 39–42).

5.2.5. Robustness. In Figure 10, we observe that as the number of black-box evaluations increases, LIME methods become more consistent, although at the cost of using more computational time. Depending on the desired robustness or time limit requirements, we can estimate of how much an explanation can change. All experiments show significant differences between them (Tables 43–46).

5.2.6. Global Conclusion. In this case, the metric of robustness is the one that outstands the most. Such results are expected since the more examples we use from the neighborhood, the less variable the generated explanation will be. Thanks to this analysis, we will be able to see what the cost is in time associated with particular robustness.

5.3. LIME vs. SHAP: General Analysis over the Explanation Generators. In this section, we evaluate the performance on each proposed metric of LLE methods, LIME with

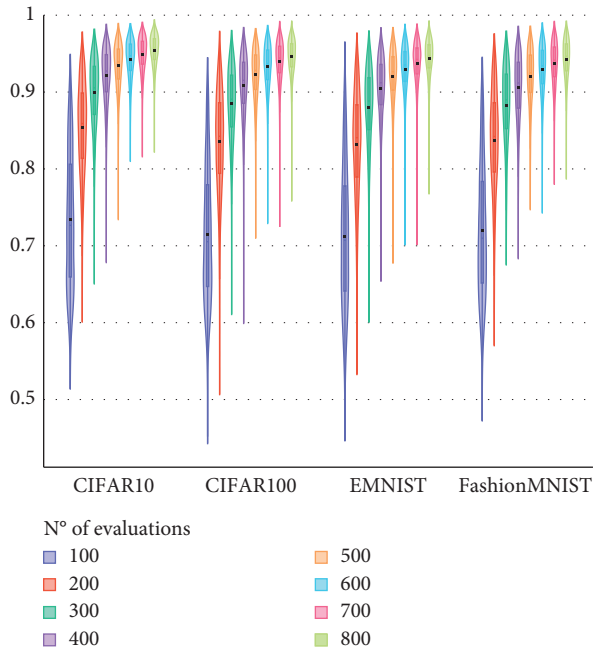

Figure 10: Performance of LIME methods grouped over max number of evaluations in the robustness metric.

Table 43: Wilcoxon test for the number of max examples and the robustness metric in the CIFAR10 dataset.

	100	200	300	400	500	600	700	800
100	—	—	—	—	—	—	—	—
200	0.0	—	—	—	—	—	—	—
300	0.0	0.0	—	—	—	—	—	—
400	0.0	0.0	0.0	—	—	—	—	—
500	0.0	0.0	0.0	0.0	—	—	—	—
600	0.0	0.0	0.0	0.0	0.0	—	—	—
700	0.0	0.0	0.0	0.0	0.0	0.0	—	—
800	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—

Table 44: Wilcoxon test for the number of max examples and the robustness metric in the CIFAR100 dataset.

	100	200	300	400	500	600	700	800
100	—	—	—	—	—	—	—	—
200	0.0	—	—	—	—	—	—	—
300	0.0	0.0	—	—	—	—	—	—
400	0.0	0.0	0.0	—	—	—	—	—
500	0.0	0.0	0.0	0.0	—	—	—	—
600	0.0	0.0	0.0	0.0	0.0	—	—	—
700	0.0	0.0	0.0	0.0	0.0	0.0	—	—
800	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—

Table 45: Wilcoxon test for the number of max examples and the robustness metric in the FashionMNIST dataset.

	100	200	300	400	500	600	700	800
100	—	—	—	—	—	—	—	—
200	0.0	—	—	—	—	—	—	—
300	0.0	0.0	—	—	—	—	—	—
400	0.0	0.0	0.0	—	—	—	—	—
500	0.0	0.0	0.0	0.0	—	—	—	—
600	0.0	0.0	0.0	0.0	0.0	—	—	—
700	0.0	0.0	0.0	0.0	0.0	0.0	—	—
800	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—

Table 46: Wilcoxon test for the number of max examples and the robustness metric in the EMNIST dataset.

	100	200	300	400	500	600	700	800
100	—	—	—	—	—	—	—	—
200	0.0	—	—	—	—	—	—	—
300	0.0	0.0	—	—	—	—	—	—
400	0.0	0.0	0.0	—	—	—	—	—
500	0.0	0.0	0.0	0.0	—	—	—	—
600	0.0	0.0	0.0	0.0	0.0	—	—	—
700	0.0	0.0	0.0	0.0	0.0	0.0	—	—
800	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—

$\sigma = 2, 3, 4, 5, 6, 7, 8$ and SHAP, local and global versions. For this comparison, we considered the results of the above scenarios to choose the best number of features and the maximum number of black-box evaluations. In our case, we pick 64 features and 800 black-box evaluations.

5.3.1. Local Concordance. In Figure 11, we show the performance of the local concordance metric over all datasets. We observe that LIME with larger σ performs worse. σ parameter controls the width of the neighborhood generated, making the original example x less relevant. On the other hand, local SHAP and global SHAP obtain stable and comparable results to those obtained by LIME with $\sigma = 2, 3, 4$ because in each SHAP regression, the relative importance of the original example x remains constant with respect to the rest of the generated neighbors. Most of the LIME experiments show significant differences between 2 experiments. However, we cannot discard that local SHAP and global SHAP behave in the same way (Tables 47–50).

5.3.2. Local Fidelity. In Figure 12, we note the same behavior for LIME methods as for the local concordance metric, i.e., the score of this metric decreases as σ is higher since the larger the neighborhood it generates, the less importance is given to the direct surroundings of the x example. We also note that SHAP methods obtain a worse result than LIME with $\sigma = 4$. This would mean that the behavior of SHAP gets worse as it moves away from the original x example. Most of the LIME experiments show significant differences between 2 experiments. However, we cannot discard that local SHAP and global SHAP behave in the same way (Tables 51–54).

5.3.3. Prescriptivity. In Figure 13, we note that different LIME methods show similar performance regardless of σ, with slight variations between datasets. On the other hand, there is a noticeable loss in SHAP local. This is partly due to the fact that SHAP gives significant weight to the original example x when there are a large number of features and does not extrapolate to more distant examples. On the other hand, global SHAP performs slightly worse than LIME methods. It pays attention not only to the closest examples to the original x example but also to the farthest possible examples. In CIFAR10 and FashionMNIST, all experiments show significant two-to-two differences. However, this is not the case in CIFAR100 or EMNIST. SHAP local and SHAP global always behave with significant differences (Tables 55–58).

5.3.4. Conciseness. In Figure 14, we note that the LIME methods have a similar behavior among the different σ configurations, obtaining slightly different results depending on the dataset. On the other hand, the global SHAP method shows worse results, which tells us that SHAP global spreads its attention over too many features. On the other hand, local SHAP obtains a comparable score with the different LIMEs, which means that both methods spread their attention over almost the same number of features. In this case, all the datasets have experiments with significant differences except CIFAR10 in the experiments with sigma greater than 4 (Tables 59–62).

5.3.5. Robustness. In Figure 15, we note that the best scoring results are obtained in this case by the SHAP models. This is due to the fact that SHAP methods choose neighbors in a stable way. LIME methods generate examples less stably as we increase the σ parameter. The reason of the increase of σ is that we also increase the size of the neighborhood and, therefore, the diversity of the generated neighbors. All experiments have significant differences with the rest of the experiments (Tables 63–66).

5.4. Global Analysis and Lessons Learned. Once we have analyzed the performance of each metric separately, we can extract lessons learned about each of the methods evaluated thanks to the auditing potential of the REVEL framework.

(i) SHAP: It focuses too much on the concrete example to be explained and does not generalize well in the synthetic neighborhood. Local concordance is good although the local fidelity, in comparison with LIME, is worse than expected and prescriptivity results are very poor. Although they are very stable methods, as we observe in the robustness metric, we may establish, in conjunction with the previous conclusions, that they are in fact methods whose neighborhood is too small and therefore they use almost all the same examples to generate explanations.

(ii) Global SHAP vs. local SHAP: The main difference between local and global SHAP is found in prescriptivity and conciseness. Local SHAP is able to discard unimportant features, while global SHAP hardly does so. The reason for this behavior is because local SHAP is using only the neighborhood

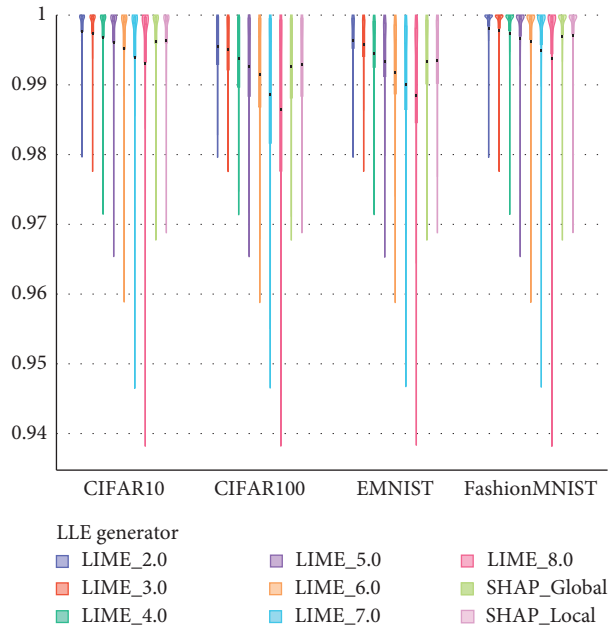

Figure 11: Performance of each explanation generator over the local concordance metric.

Table 47: Wilcoxon test for the different XAI methods and the local concordance metric in the CIFAR10 dataset.

	LIME 2	LIME 3	LIME 4	LIME 5	LIME 6	LIME 7	LIME 8	SHAP global	SHAP local
LIME 2	—	—	—	—	—	—	—	—	—
LIME 3	0.224211	—	—	—	—	—	—	—	—
LIME 4	0.0	0.000045	—	—	—	—	—	—	—
LIME 5	0.0	0.0	0.001585	—	—	—	—	—	—
LIME 6	0.0	0.0	0.0	0.004809	—	—	—	—	—
LIME 7	0.0	0.0	0.0	0.0	0.000511	—	—	—	—
LIME 8	0.0	0.0	0.0	0.0	0.0	0.096012	—	—	—
SHAP global	0.0	0.0	0.038764	0.158645	0.00001	0.0	0.0	—	—
SHAP local	0.0	0.0	0.346758	0.016339	0.000002	0.0	0.0	0.168907	—

Table 48: Wilcoxon test for the different XAI methods and the local concordance metric in the CIFAR100 dataset.

	LIME 2	LIME 3	LIME 4	LIME 5	LIME 6	LIME 7	LIME 8	SHAP global	SHAP local
LIME 2	—	—	—	—	—	—	—	—	—
LIME 3	0.296419	—	—	—	—	—	—	—	—
LIME 4	0.0	0.0	—	—	—	—	—	—	—
LIME 5	0.0	0.0	0.000032	—	—	—	—	—	—
LIME 6	0.0	0.0	0.0	0.036054	—	—	—	—	—
LIME 7	0.0	0.0	0.0	0.0	0.00005	—	—	—	—
LIME 8	0.0	0.0	0.0	0.0	0.0	0.001009	—	—	—
SHAP global	0.0	0.000002	0.248288	0.0	0.0	0.0	0.0	—	—
SHAP local	0.0	0.000388	0.025539	0.0	0.0	0.0	0.0	0.381647	—

near the instance to be analyzed, while global SHAP uses also the instances of completely empty images except for some particular patches. In other types of data, this approach is correct (e.g., in tabular data, to see if any particular feature biases the overall result), but in the case of images, an almost entirely grey image does not give much information.

(iii) LIME: This method focuses on the local neighborhood of the example to be explained. We observe that the parameter σ establishes the size of the neighborhood, and as it increases, it obtains worse results in the local environment but has greater generalization power. We deduce this because in the metrics of local concordance and local fidelity, it

Table 49: Wilcoxon test for the different XAI methods and the local concordance metric in the FashionMNIST dataset.

	LIME 2	LIME 3	LIME 4	LIME 5	LIME 6	LIME 7	LIME 8	SHAP global	SHAP local
LIME 2	—	—	—	—	—	—	—	—	—
LIME 3	0.036628	—	—	—	—	—	—	—	—
LIME 4	0.00007	0.036533	—	—	—	—	—	—	—
LIME 5	0.0	0.0	0.000029	—	—	—	—	—	—
LIME 6	0.0	0.0	0.0	0.004669	—	—	—	—	—
LIME 7	0.0	0.0	0.0	0.0	0.000001	—	—	—	—
LIME 8	0.0	0.0	0.0	0.0	0.0	0.000006	—	—	—
SHAP global	0.0	0.000001	0.00191	0.221104	0.001733	0.0	0.0	—	—
SHAP local	0.0	0.0	0.00107	0.647406	0.000414	0.0	0.0	0.656478	—

Table 50: Wilcoxon test for the different XAI methods and the local concordance metric in the EMNIST dataset.

	LIME 2	LIME 3	LIME 4	LIME 5	LIME 6	LIME 7	LIME 8	SHAP global	SHAP local
LIME 2	—	—	—	—	—	—	—	—	—
LIME 3	0.0	—	—	—	—	—	—	—	—
LIME 4	0.0	0.0	—	—	—	—	—	—	—
LIME 5	0.0	0.0	0.0	—	—	—	—	—	—
LIME 6	0.0	0.0	0.0	0.0	—	—	—	—	—
LIME 7	0.0	0.0	0.0	0.0	0.0	—	—	—	—
LIME 8	0.0	0.0	0.0	0.0	0.0	0.0	—	—	—
SHAP global	0.0	0.0	0.0	0.125947	0.0	0.0	0.0	—	—
SHAP local	0.0	0.0	0.0	0.205828	0.0	0.0	0.0	0.928104	—

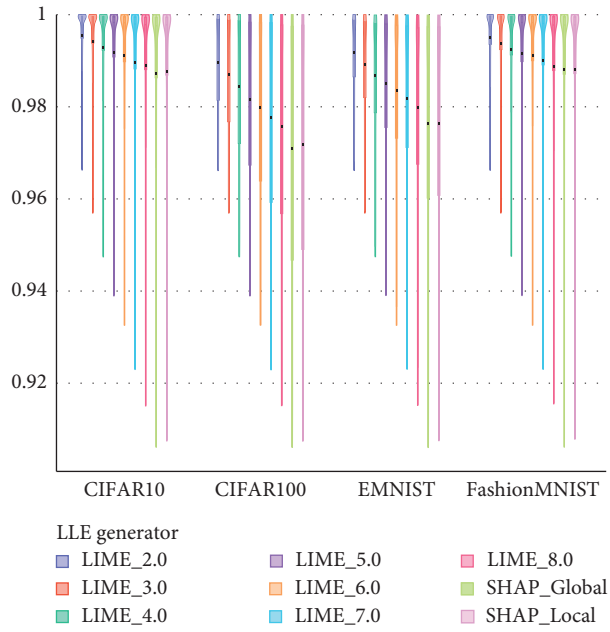

Figure 12: Performance of each explanation generator over the local fidelity metric.

worsens with increasing σ but remains stable or even increases in prescriptivity. The increase in neighborhood size also results in slightly more attention being paid to diverse features and, in addition, causes a more diverse generation of neighbors, as we see in the conciseness and robustness metrics, respectively.

In conclusion, we may establish that SHAP focuses too much on the example to be explained while LIME is able to generalize better on these datasets.

Finally, the most important lesson learned is the exhaustive and mathematically robust study we performed for the development of REVEL. Thanks to this study, we have not only been able to establish comparative measures

Table 51: Wilcoxon test for the different XAI methods and the local fidelity metric in the CIFAR10 dataset.

	LIME 2	LIME 3	LIME 4	LIME 5	LIME 6	LIME 7	LIME 8	SHAP global	SHAP local
LIME 2	—	—	—	—	—	—	—	—	—
LIME 3	0.000016	—	—	—	—	—	—	—	—
LIME 4	0.0	0.000051	—	—	—	—	—	—	—
LIME 5	0.0	0.0	0.002586	—	—	—	—	—	—
LIME 6	0.0	0.0	0.0	0.053614	—	—	—	—	—
LIME 7	0.0	0.0	0.0	0.0	0.002317	—	—	—	—
LIME 8	0.0	0.0	0.0	0.0	0.0	0.11511	—	—	—
SHAP global	0.0	0.0	0.0	0.0	0.000017	0.466122	0.331583	—	—
SHAP local	0.0	0.0	0.0	0.0	0.000652	0.336936	0.031794	0.420162	—

Table 52: Wilcoxon test for the different XAI methods and the local fidelity metric in the CIFAR100 dataset.

	LIME 2	LIME 3	LIME 4	LIME 5	LIME 6	LIME 7	LIME 8	SHAP global	SHAP local
LIME 2	—	—	—	—	—	—	—	—	—
LIME 3	0.0	—	—	—	—	—	—	—	—
LIME 4	0.0	0.0	—	—	—	—	—	—	—
LIME 5	0.0	0.0	0.0	—	—	—	—	—	—
LIME 6	0.0	0.0	0.0	0.001361	—	—	—	—	—
LIME 7	0.0	0.0	0.0	0.0	0.000001	—	—	—	—
LIME 8	0.0	0.0	0.0	0.0	0.0	0.000072	—	—	—
SHAP global	0.0	0.0	0.0	0.0	0.0	0.0	0.32619	—	—
SHAP local	0.0	0.0	0.0	0.0	0.0	0.000001	0.590278	0.780151	—

Table 53: Wilcoxon test for the different XAI methods and the local fidelity metric in the FashionMNIST dataset.

	LIME 2	LIME 3	LIME 4	LIME 5	LIME 6	LIME 7	LIME 8	SHAP global	SHAP local
LIME 2	—	—	—	—	—	—	—	—	—
LIME 3	0.0	—	—	—	—	—	—	—	—
LIME 4	0.0	0.008124	—	—	—	—	—	—	—
LIME 5	0.0	0.0	0.001078	—	—	—	—	—	—
LIME 6	0.0	0.0	0.000013	0.153165	—	—	—	—	—
LIME 7	0.0	0.0	0.0	0.000005	0.00009	—	—	—	—
LIME 8	0.0	0.0	0.0	0.0	0.0	0.001501	—	—	—
SHAP global	0.0	0.0	0.0	0.0	0.0	0.050607	0.14538	—	—
SHAP local	0.0	0.0	0.0	0.0	0.0	0.007146	0.476541	0.245657	—

Table 54: Wilcoxon test for the different XAI methods and the local fidelity metric in the EMNIST dataset.

	LIME 2	LIME 3	LIME 4	LIME 5	LIME 6	LIME 7	LIME 8	SHAP global	SHAP local
LIME 2	—	—	—	—	—	—	—	—	—
LIME 3	0.0	—	—	—	—	—	—	—	—
LIME 4	0.0	0.0	—	—	—	—	—	—	—
LIME 5	0.0	0.0	0.0	—	—	—	—	—	—
LIME 6	0.0	0.0	0.0	0.0	—	—	—	—	—
LIME 7	0.0	0.0	0.0	0.0	0.0	—	—	—	—
LIME 8	0.0	0.0	0.0	0.0	0.0	0.0	—	—	—
SHAP global	0.0	0.0	0.0	0.0	0.0	0.0	0.024961	—	—
SHAP local	0.0	0.0	0.0	0.0	0.0	0.0	0.037695	0.402308	—

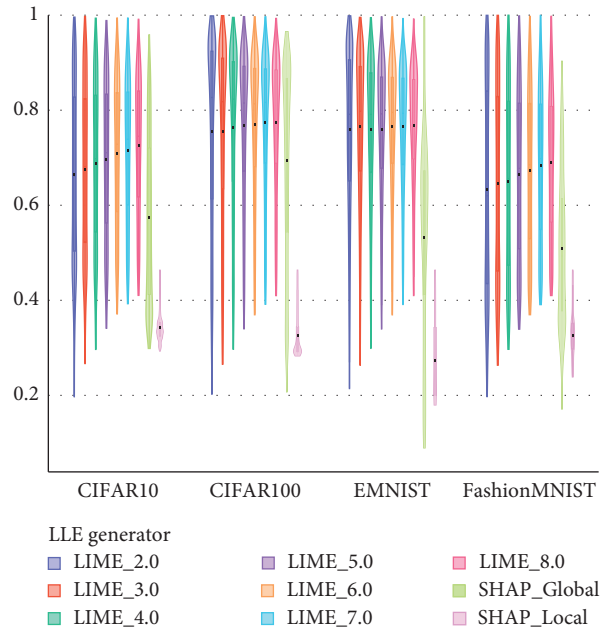

Figure 13: Performance of each explanation generator over the prescriptivity metric.

Table 55: Wilcoxon test for the different XAI methods and the prescriptivity metric in the CIFAR10 dataset

	LIME 2	LIME 3	LIME 4	LIME 5	LIME 6	LIME 7	LIME 8	SHAP global	SHAP local
LIME 2	—	—	—	—	—	—	—	—	—
LIME 3	0.003363	—	—	—	—	—	—	—	—
LIME 4	0.0	0.000565	—	—	—	—	—	—	—
LIME 5	0.0	0.0	0.046085	—	—	—	—	—	—
LIME 6	0.0	0.0	0.0	0.000023	—	—	—	—	—
LIME 7	0.0	0.0	0.0	0.0	0.029132	—	—	—	—
LIME 8	0.0	0.0	0.0	0.0	0.000012	0.024575	—	—	—
SHAP global	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—	—
SHAP local	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—

Table 56: Wilcoxon test for the different XAI methods and the prescriptivity metric in the CIFAR100 dataset.

	LIME 2	LIME 3	LIME 4	LIME 5	LIME 6	LIME 7	LIME 8	SHAP global	SHAP local
LIME 2	—	—	—	—	—	—	—	—	—
LIME 3	0.241746	—	—	—	—	—	—	—	—
LIME 4	0.543763	0.429915	—	—	—	—	—	—	—
LIME 5	0.72744	0.90477	0.63844	—	—	—	—	—	—
LIME 6	0.310117	0.944617	0.514841	0.780728	—	—	—	—	—
LIME 7	0.629185	0.637106	0.808105	0.765539	0.626447	—	—	—	—
LIME 8	0.227314	0.655869	0.2653	0.301112	0.598209	0.290457	—	—	—
SHAP global	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—	—
SHAP local	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—

Table 57: Wilcoxon test for the different XAI methods and the prescriptivity metric in the FashionMNIST dataset.

	LIME 2	LIME 3	LIME 4	LIME 5	LIME 6	LIME 7	LIME 8	SHAP global	SHAP local
LIME 2	—	—	—	—	—	—	—	—	—
LIME 3	0.004948	—	—	—	—	—	—	—	—
LIME 4	0.004952	0.870345	—	—	—	—	—	—	—
LIME 5	0.0	0.002011	0.001656	—	—	—	—	—	—
LIME 6	0.0	0.000061	0.000006	0.08854	—	—	—	—	—
LIME 7	0.0	0.0	0.0	0.000057	0.009363	—	—	—	—
LIME 8	0.0	0.0	0.0	0.0	0.000352	0.422926	—	—	—
SHAP global	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—	—
SHAP local	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—

Table 58: Wilcoxon test for the different XAI methods and the prescriptivity metric in the EMNIST dataset.

	LIME 2	LIME 3	LIME 4	LIME 5	LIME 6	LIME 7	LIME 8	SHAP global	SHAP local
LIME 2	—	—	—	—	—	—	—	—	—
LIME 3	0.481407	—	—	—	—	—	—	—	—
LIME 4	0.275306	0.017817	—	—	—	—	—	—	—
LIME 5	0.04566	0.000502	0.386528	—	—	—	—	—	—
LIME 6	0.240065	0.081447	0.839167	0.201618	—	—	—	—	—
LIME 7	0.060687	0.009952	0.773493	0.848934	0.471	—	—	—	—
LIME 8	0.479104	0.049914	0.920686	0.113797	0.922287	0.211443	—	—	—
SHAP global	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—	—
SHAP local	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—

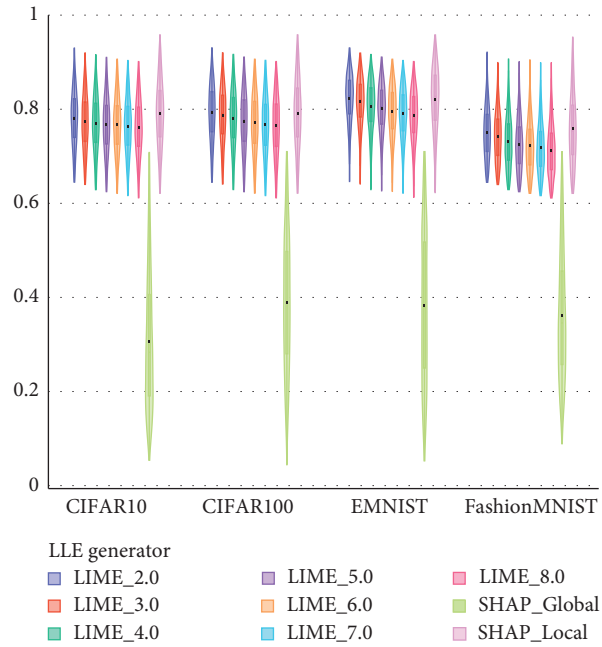

Figure 14: Performance of each explanation generator over the conciseness metric.

Table 59: Wilcoxon test for the different XAI methods and the conciseness metric in the CIFAR10 dataset.

	LIME 2	LIME 3	LIME 4	LIME 5	LIME 6	LIME 7	LIME 8	SHAP global	SHAP local
LIME 2	—	—	—	—	—	—	—	—	—
LIME 3	0.0	—	—	—	—	—	—	—	—
LIME 4	0.0	0.024372	—	—	—	—	—	—	—
LIME 5	0.0	0.000042	0.017442	—	—	—	—	—	—
LIME 6	0.0	0.000921	0.119171	0.182328	—	—	—	—	—
LIME 7	0.0	0.0	0.000891	0.206938	0.012491	—	—	—	—
LIME 8	0.0	0.0	0.000001	0.006758	0.000132	0.239556	—	—	—
SHAP global	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—	—
SHAP local	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—

Table 60: Wilcoxon test for the different XAI methods and the conciseness metric in the CIFAR100 dataset.

	LIME 2	LIME 3	LIME 4	LIME 5	LIME 6	LIME 7	LIME 8	SHAP global	SHAP local
LIME 2	—	—	—	—	—	—	—	—	—
LIME 3	0.000093	—	—	—	—	—	—	—	—
LIME 4	0.0	0.0	—	—	—	—	—	—	—
LIME 5	0.0	0.0	0.000002	—	—	—	—	—	—
LIME 6	0.0	0.0	0.0	0.013876	—	—	—	—	—
LIME 7	0.0	0.0	0.0	0.0	0.000216	—	—	—	—
LIME 8	0.0	0.0	0.0	0.0	0.0	0.044479	—	—	—
SHAP global	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—	—
SHAP local	0.412729	0.011452	0.0	0.0	0.0	0.0	0.0	0.0	—

Table 61: Wilcoxon test for the different XAI methods and the conciseness metric in the FashionMNIST dataset.

	LIME 2	LIME 3	LIME 4	LIME 5	LIME 6	LIME 7	LIME 8	SHAP global	SHAP local
LIME 2	—	—	—	—	—	—	—	—	—
LIME 3	0.0	—	—	—	—	—	—	—	—
LIME 4	0.0	0.0	—	—	—	—	—	—	—
LIME 5	0.0	0.0	0.000002	—	—	—	—	—	—
LIME 6	0.0	0.0	0.0	0.000679	—	—	—	—	—
LIME 7	0.0	0.0	0.0	0.0	0.002381	—	—	—	—
LIME 8	0.0	0.0	0.0	0.0	0.0	0.000381	—	—	—
SHAP global	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—	—
SHAP local	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—

Table 62: Wilcoxon test for the different XAI methods and the conciseness metric in the EMNIST dataset.

	LIME 2	LIME 3	LIME 4	LIME 5	LIME 6	LIME 7	LIME 8	SHAP global	SHAP local
LIME 2	—	—	—	—	—	—	—	—	—
LIME 3	0.0	—	—	—	—	—	—	—	—
LIME 4	0.0	0.0	—	—	—	—	—	—	—
LIME 5	0.0	0.0	0.0	—	—	—	—	—	—
LIME 6	0.0	0.0	0.0	0.0	—	—	—	—	—
LIME 7	0.0	0.0	0.0	0.0	0.000011	—	—	—	—
LIME 8	0.0	0.0	0.0	0.0	0.0	0.0	—	—	—
SHAP global	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—	—
SHAP local	0.674696	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—

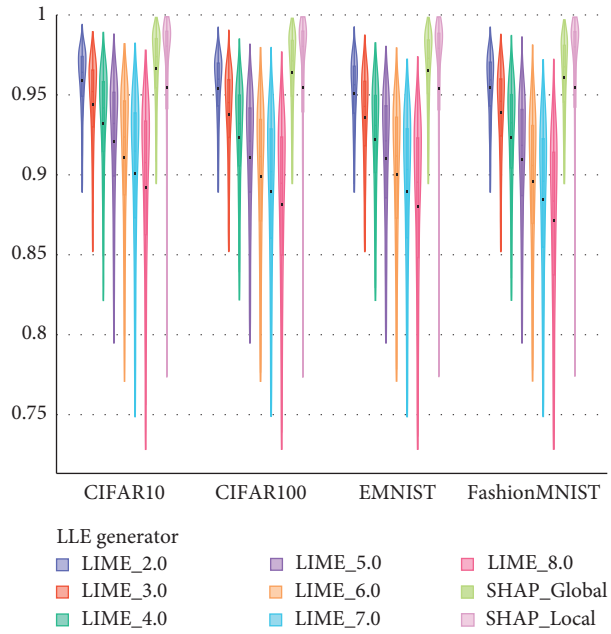

Figure 15: Performance of each explanation generator over the robustness metric.

Table 63: Wilcoxon test for the different XAI methods and the robustness metric in the CIFAR10 dataset.

	LIME 2	LIME 3	LIME 4	LIME 5	LIME 6	LIME 7	LIME 8	SHAP global	SHAP local
LIME 2	—	—	—	—	—	—	—	—	—
LIME 3	0.0	—	—	—	—	—	—	—	—
LIME 4	0.0	0.0	—	—	—	—	—	—	—
LIME 5	0.0	0.0	0.0	—	—	—	—	—	—
LIME 6	0.0	0.0	0.0	0.0	—	—	—	—	—
LIME 7	0.0	0.0	0.0	0.0	0.0	—	—	—	—
LIME 8	0.0	0.0	0.0	0.0	0.0	0.0	—	—	—
SHAP global	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—	—
SHAP local	0.0	0.000685	0.000001	0.0	0.0	0.0	0.0	0.0	—

Table 64: Wilcoxon test for the different XAI methods and the robustness metric in the CIFAR100 dataset.

	LIME 2	LIME 3	LIME 4	LIME 5	LIME 6	LIME 7	LIME 8	SHAP global	SHAP local
LIME 2	—	—	—	—	—	—	—	—	—
LIME 3	0.0	—	—	—	—	—	—	—	—
LIME 4	0.0	0.0	—	—	—	—	—	—	—
LIME 5	0.0	0.0	0.0	—	—	—	—	—	—
LIME 6	0.0	0.0	0.0	0.0	—	—	—	—	—
LIME 7	0.0	0.0	0.0	0.0	0.0	—	—	—	—
LIME 8	0.0	0.0	0.0	0.0	0.0	0.0	—	—	—
SHAP global	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—	—
SHAP local	0.0	0.058395	0.0	0.0	0.0	0.0	0.0	0.0	—

Table 65: Wilcoxon test for the different XAI methods and the robustness metric in the FashionMNIST dataset.

	LIME 2	LIME 3	LIME 4	LIME 5	LIME 6	LIME 7	LIME 8	SHAP global	SHAP local
LIME 2	—	—	—	—	—	—	—	—	—
LIME 3	0.0	—	—	—	—	—	—	—	—
LIME 4	0.0	0.0	—	—	—	—	—	—	—
LIME 5	0.0	0.0	0.0	—	—	—	—	—	—
LIME 6	0.0	0.0	0.0	0.0	—	—	—	—	—
LIME 7	0.0	0.0	0.0	0.0	0.0	—	—	—	—
LIME 8	0.0	0.0	0.0	0.0	0.0	0.0	—	—	—
SHAP global	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—	—
SHAP local	0.0	0.536291	0.0	0.0	0.0	0.0	0.0	0.0	—

Table 66: Wilcoxon test for the different XAI methods and the robustness metric in the EMNIST dataset.

	LIME 2	LIME 3	LIME 4	LIME 5	LIME 6	LIME 7	LIME 8	SHAP global	SHAP local
LIME 2	—	—	—	—	—	—	—	—	—
LIME 3	0.0	—	—	—	—	—	—	—	—
LIME 4	0.0	0.0	—	—	—	—	—	—	—
LIME 5	0.0	0.0	0.0	—	—	—	—	—	—
LIME 6	0.0	0.0	0.0	0.0	—	—	—	—	—
LIME 7	0.0	0.0	0.0	0.0	0.0	—	—	—	—
LIME 8	0.0	0.0	0.0	0.0	0.0	0.0	—	—	—
SHAP global	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—	—
SHAP local	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—

between explanations but also show that these measures serve as absolute measures, without the need to compare with others.

6. Concluding Remarks

In this paper, we present REVEL, a novel framework specialized in analysis and comparison of explanations. We provide a theoretical guideline for the use of REVEL. We also provide a practical illustration of usage of REVEL by comparing LIME and SHAP methods in four different benchmarks.

As lessons learned, we want to remark that having bounded metrics with well-defined limits gives us absolute information on every evaluation aspect and not only a comparative one. This is useful to dismiss explanations by themselves even if there is no baseline to compare with. For the development of future metrics, this characteristic is desirable.

Regarding the developed metrics themselves, we can extract the following lessons. Local metrics can help us to detect biases; compared with prescriptivity, conciseness provides us information about whether an explanation is useful or not by the percentage of discarded features, and robustness shows information on the stability of the explanations.

From the above analysis, we can establish that, within the black-box methods of explanation proposals over the image classification task, LIME behaves better than SHAP because SHAP focuses too much on the locality of the example to be explained, while LIME is able to generalize much better.

Once the method of explanation has been chosen for a particular model, we emphasize that the analysis should not stop there but analyze different aspects such as the number of features considered or the number of evaluations of the black box necessary for a robust explanation.

Finally, we consider that the base case on which to work rigorously in XAI is LLE. As future work, and based on the study already done, we leave the extension of the proposed metrics to other types of explanations, such as those based on decision trees or knowledge graphs.

Data Availability

The image data supporting this study are from previously reported studies and datasets, which have been cited. Datasets are available from the torchvision datasets library.

Disclosure

This paper is based on a preprint "REVEL Framework to Measure Local Linear Explanations for Black-Box Models: Deep Learning Image Classification Case of Study" in 2022 based on the following link: https://arxiv.org/abs/2211.06154 [8].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Spanish Ministry of Science and Technology under project PID2020-119478GB-I00 financed by \\ MCIN/AEI/10.13039/501100011033. This work was also partially supported by the Contract UGR-AM OTRI-6717 and the Contract UGR-AM OTRI-5987 and projects P18-FR-4961 by Proyectos I+D+i Junta de Andalucia 2018. The hardware used in this work is supported by the projects with reference EQC2018-005084-P granted by Spain's Ministry of Science and Innovation and European Regional Development Fund (ERDF) and the project with reference SOMM17/6110/UGR granted by the Andalusian "Consejería de Conocimiento, Investigación y Universidades" and European Regional Development Fund (ERDF).

References

[1] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser et al., "Explainable artificial intelligence (xai): concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[2] A. Yan, R. Hou, X. Liu, H. Yan, T. Huang, and X. Wang, "Towards explainable model extraction attacks," *International Journal of Intelligent Systems*, vol. 37, no. 11, pp. 9936–9956, 2022.

[3] M. Bohanec, M. K. Borštnar, and M. Robnik-Šikonja, "Explaining machine learning models in sales predictions," *Expert Systems with Applications*, vol. 71, pp. 416–428, 2017.

[4] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Computing Surveys*, vol. 51, no. 5, pp. 1–42, 2018.

[5] R. Confalonieri, L. Coba, B. Wagner, and T. R. Besold, "A historical perspective of explainable artificial intelligence," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 11, no. 1, p. e1391, 2021.

[6] E. Amparore, A. Perotti, and P. Bajardi, "To trust or not to trust an explanation: using leaf to evaluate local linear xai methods," *PeerJ Computer Science*, vol. 7, p. e479, 2021.

[7] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: an overview of interpretability of machine learning," in *Proceedings of the 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 80–89, IEEE, Turin, Italy, October, 2018.

[8] I. Sevillano-García, J. Luengo-Martín, and F. Herrera, "Revel framework to measure local linear explanations for black-box models: Deep learning image classification case of study," 2022, https://arxiv.org/abs/2211.06154.

[9] D. Slack, A. Hilgard, S. Singh, and H. Lakkaraju, "Reliable post hoc explanations: modeling uncertainty in explainability," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34, pp. 9391–9404, Curran Associates, Inc, New York, NY, USA, 2021.

[10] M. T. Ribeiro, S. Singh, and C. 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, KDD '16*, pp. 1135–1144, Association for Computing Machinery, New York, NY, USA, October, 2016.

[11] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, I. Guyon, S. Luxburg, H. Bengio, R. F. Wallach, S. Vishwanathan, and R. Garnett, Eds., vol. 30, pp. 4765–4774, Curran Associates, Inc, New York, NY, USA, 2017.

[12] P. Zhu and M. Ogino, "Guideline-based additive explanation for computer-aided diagnosis of lung nodules," in *Interpretability of Machine Intelligence in Medical Image Computing and Multimodal Learning for Clinical Decision Support*, pp. 39–47, Springer, Berlin, Germany, 2019.

[13] L. Schallner, J. Rabold, O. Scholz, and U. Schmid, "Effect of superpixel aggregation on explanations in lime–a case study with biological data," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 147–158, Springer, Berlin, Germany, 2019.

[14] O. Loyola-González, A. E. Gutierrez-Rodríguez, M. A. Medina-Pérez et al., "An explainable artificial intelligence model for clustering numerical databases," *IEEE Access*, vol. 8, pp. 52370–52384, 2020.

[15] A. Rosenfeld, "Better metrics for evaluating explainable artificial intelligence," in *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems*, pp. 45–50, London, UK, May, 2021.

[16] C. Gentile and M. K. Warmuth, "Linear hinge loss and average margin," *Advances in Neural Information Processing Systems*, vol. 11, pp. 225–231, 1998.

[17] A. Krizhevsky, V. Nair, and G. Hinton, "Cifar-10 (canadian institute for advanced research)," vol. 5, no. 4, 2010.

[18] A. Krizhevsky, *Learning Multiple Layers of Features From Tiny Images*, 2009.

[19] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," 2017, https://arxiv.org/abs/1708.07747.

[20] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "Emnist: extending mnist to handwritten letters," in *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926, IEEE, Anchorage, AL, USA, May, 2017.

[21] M. Tan and Q. Le, "Efficientnet: rethinking model scaling for convolutional neural networks," in *Proceedings of the International Conference on Machine Learning*, pp. 6105–6114, Montreal, Canada, June, 2019.

[22] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.

[23] P. Virtanen, R. Gommers, T. E. Oliphant et al., "SciPy 1.0: fundamental algorithms for scientific computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.