

c-Eval: A Unified Metric to Evaluate Feature-based Explanations via Perturbation

Minh N. Vu* Truc D. Nguyen* NhatHai Phan[†] Ralucca Gera[‡] My T. Thai*

* University of Florida, Gainesville, Florida, USA

[†] New Jersey Institute of Technology, Newark, New Jersey, USA

[‡] Naval Postgraduate School, Monterey, California, USA

minhvu@ufl.edu, truc.nguyen@ufl.edu, phan@njit.edu, rgera@nps.edu, mythai@cise.ufl.edu

Abstract—In many image-classification applications, understanding the reasons of model's prediction can be as critical as the prediction's accuracy itself. Various feature-based local explainers have been designed to provide explanations on the decision of complex classifiers. Nevertheless, there is no consensus on evaluating the quality of different explanations. In response to this lack of comprehensive evaluation, we introduce the c-Eval metric and its corresponding framework to quantify the feature-based local explanation's quality. Given a classifier's prediction and the corresponding explanation on that prediction, c-Eval is the minimum-distortion perturbation that successfully alters the prediction while keeping the explanation's features unchanged. To show that c-Eval captures the importance of input's features, we establish a connection between c-Eval and the features returned by explainers in affine and nearly-affine classifiers. We then introduce the c-Eval plot, which not only displays a strong connection between c-Eval and explainers' quality, but also helps automatically determine explainer's parameters.

Index Terms—Explainable/Interpretable Machine Learning, Feature-based Local Explainers, Metric, Image Classification.

I. INTRODUCTION

With the pervasiveness of machine learning in many emerging domains, especially in critical applications such as health-care or autonomous systems, it is utmost important to understand why a machine learning model makes such a prediction. For example, deep convolutional neural networks (CNNs) have been deployed to detect skin cancer at a level of competence comparable to dermatologists [1]. However, doctors and experts cannot rely on these predictions blindly. Providing additional intelligible explanations such as a highlighted skin region that contributes to the prediction will aid doctors significantly in making their diagnoses. Along this direction, many machine learning explainers supporting users in interpreting the predictions of complex neural networks on given inputs, called *local explainers*, have been proposed and studied, such as SHAP [2], LIME [3], Grad-CAM (GCam) [4], and DeepLIFT [5], among others [6]–[13]. Since the output of these explainers, called *explanation*, is either a subset or weights of the input's features, these explainers are often referred as feature-based local explainers.

Even though it is important to evaluate the explanations, the evaluation task remains challenging [14], [15]. One major

challenge is the lack of ground-truth explanations. In fact, most feature-based explanations have been evaluated only through a small set of human-based experiments which apparently does not imply the global guarantee on their quality [2], [5]. Another challenge is the diversity in the presentation/format of explanations. Fig. 1 shows an example of three explanations generated by LIME, GCam, and SHAP explainers for the prediction *Pembroke* made by the Inception-v3 image classifier [16]. All of them highlight the region containing the *Pembroke*; however, their formats vary from picture segments in LIME, heat-map in GCam to pixel importance-weights in SHAP. Furthermore, explainers might be designed for different objectives as there is a fundamental trade-off between the interpretability and the accuracy of explanations [2], [3]. In fact, utilizing for the interpretability of explanation might cost its consistency with the explained prediction. The diversity in presentations and the difference in objectives are great roadblocks to the evaluation of explanations.

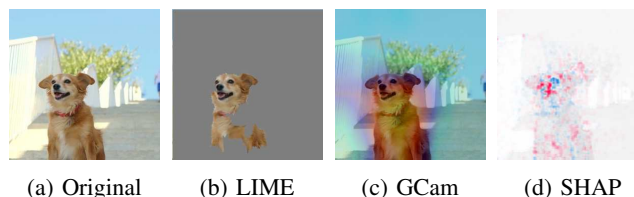


Fig. 1: Explanations generated by three feature-based local explainers for the prediction *Pembroke* of Inception-v3 model.

Contribution. Our work focuses on evaluating explanations of feature-based local explainers. Specifically, we evaluate the power of the explanations toward the model's prediction, i.e. if only the features including in the explanation are observed, how certain the model is on the explained prediction. We first introduce a novel metric, c-Eval, to evaluate the quality of explanations. The metric is based on the intuition that if certain features are important to the prediction, it is difficult to change the prediction when those features are kept intact. The power of an explanation is therefore quantified by the minimum amount of perturbation on features outside of the explanation that can alter the prediction.

We further provide analysis showing a connection between

Corresponding author: My T. Thai

the importance of features containing in an explanation and its corresponding c -Eval in multi-class affine classifiers. For general non-affine classifiers, our experimental results based on c -Eval suggest an existence of nearly-affine decision surfaces in many modern classifiers. This observation encourages an adoption of the c -Eval metric in evaluating explanations of predictions made by a broad range of image classifiers.

Additionally, we introduce the c -Eval plot, an approach based on c -Eval to visualize explainers' behaviors on a given input. Using LIME explainer as an example, we show how c -Eval plot helps us gain more trust on LIME and select appropriate parameters for the explainers. We also heuristically demonstrate the behaviors of c -Eval in adversarial-robust models. Our results show that the c -Eval computed in robust models is highly correlated with the non-robust counterpart, which strengthens and validates the applications of c -Eval.

Related Work. Despite the recent development of explainable AI, works focusing on evaluating explanations of local feature-based explainers are quite limited. To our knowledge, there are two works that can be considered to be relevant to c -Eval: the AOPC score [17] and the log-odds score [5]. The AOPC score, which is introduced to evaluate heat-maps, is the average of the differences between the soft-outputs of the input image and those of some random perturbations. These random perturbations are generated sequentially based on the heat-maps on the input's features. One may think to extend AOPC to evaluate explainers, such as mask-form explanation LIME; however, it is ambiguous due to an absence of the importance ordering. Furthermore, the AOPC requires a large number of random perturbations to generate stable evaluations while computing c -Eval is a deterministic process requiring only one perturbation per evaluation. On the other hand, Shrikumar *et al* [5] use the log-odds score, measuring the difference between the input image and the modified image whose some pixels are erased, to evaluate explanations [5]. In this measurement, the erased pixels are chosen greedily based on the importance-weights given by the explanation. Then, the explanation is evaluated based on how many erased pixels are needed to alter the original predicted label. However, the log-odds method is proposed without detailed analysis and it is only applicable to small gray-scale images, such as MNIST [18].

Organization. The rest of the paper is organized as follows. Section II introduces notations and formulates c -Eval. Section III shows how to compute c -Eval. The relationship between c -Eval and the importance of input features is demonstrated in Section IV. The c -Eval plot, a visualization method based on c -Eval to examine explainers' behavior, is presented in Section V-A. Section V-B includes our demonstration of c -Eval on adversarial-robust models. Our experimental evaluations on explanations to validate the usage of c -Eval are demonstrated in Section VI. Finally, Section VII concludes the paper with a discussion on future directions.

II. C-EVAL OF EXPLANATION

In this section, we introduce our notions and describe the formulation of the c -Eval metric. We model a neural network

as a function f whose input and output are vectors $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$. For a given vector $\mathbf{x}$, x_i is the element i^{th} of $\mathbf{x}$. The predicted label of the model's prediction $\mathbf{y} = f(\mathbf{x})$ is $l = \arg \max_{1 \leq j \leq m} y_j$. Given g_f , a feature-based local explainer on the classifier f , an explanation of prediction $f(\mathbf{x})$ is a subset of features/elements of $\mathbf{x}$, i.e. $e_{\mathbf{x}} = g_f(\mathbf{x}) \subseteq \mathbf{x}$. We call $e_{\mathbf{x}}$ the *explanatory features* and $\mathbf{x} \setminus e_{\mathbf{x}}$ the *non-explanatory features* of prediction $f(\mathbf{x})$ generated by g_f .

In feature-based explanations, the explainer may simply return $e_{\mathbf{x}} = \mathbf{x}$ as an explanation for prediction $f(\mathbf{x})$. We can interpret this naive explanation as *because the input is $\mathbf{x}$ so the prediction is $f(\mathbf{x})$* . Note that this explanation is not desirable since it neither gives us any additional information on the prediction nor strengthens our trust on the model. A better answer is a smaller set of explanatory features that are important to the prediction. Thus, it is a common practice for explainers to impose cardinality constraints on $e_{\mathbf{x}}$ for more compact explanations [3], [13]. When evaluating the explanations, we assume that they are all subjected to the same cardinality constraint $|e_{\mathbf{x}}| \leq k$ for a fix given integer k .

We denote a perturbation scheme $h_{g_f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ with respect to (w.r.t) explainer g_f to be a function from the input's space $\mathbb{R}^n$ to itself. The resulted perturbation $h_{g_f}(\mathbf{x})$ is only allowed to be different from $\mathbf{x}$ on non-explanatory features of $f(\mathbf{x})$, i.e. $\mathbf{x} \setminus e_{\mathbf{x}}$:

$$h_{g_f}(\mathbf{x})_i = \begin{cases} x_i & \text{if } x_i \in e_{\mathbf{x}} \\ x_i + \delta_i & \text{if } x_i \notin e_{\mathbf{x}}, \end{cases} \quad (1)$$

where $\delta_i \in \mathbb{R}$ is the perturbation on component i^{th} . A perturbation $h_{g_f}(\mathbf{x})$ is considered successful under the p -norm constraint c if the predicted label of the model on $h_{g_f}(\mathbf{x})$ is different from the original predicted label on $\mathbf{x}$ and the p -norm difference between $\mathbf{x}$ and $h_{g_f}(\mathbf{x})$ is bounded by c :

$$\begin{aligned} \arg \max_{1 \leq j \leq m} f(h_{g_f}(\mathbf{x})) &\neq l \\ \|h_{g_f}(\mathbf{x}) - \mathbf{x}\|_p &\leq c. \end{aligned} \quad (2)$$

We denote this p -norm difference between $\mathbf{x}$ and $h_{g_f}(\mathbf{x})$ as the *perturbing distortion*. c -Eval is then defined as follows:

Definition 1. An explainer g_f (or the corresponding explanation $e_{\mathbf{x}}$) of a prediction $f(\mathbf{x})$ is c -Eval if no perturbing scheme h_{g_f} can change the model prediction on $\mathbf{x}$ while keeping the perturbing distortion less than or equal to c .

Based on Definition 1, a good feature-based explanation is supposed to be c -Eval with large c . Because, if the explanatory features $e_{\mathbf{x}}$ had high power to the prediction $f(\mathbf{x})$, modifying values on the non-explanatory features $\mathbf{x} \setminus e_{\mathbf{x}}$ have negligible impact to the model's prediction. As the perturbation scheme h_{g_f} is only on non-explanatory features, h_{g_f} must make significant modifications to successfully alter the predicted label. Consequently, for a given explanation $e_{\mathbf{x}}$, the greatest value of c (in eq. (2)) such that there is no h_{g_f} can successfully

change the predicted label would imply the power of features in e_x . Thus, we denote:

$$c_{f,x}(e_x) = \sup c \quad \text{s.t. } \nexists h_{g_f} \text{ satisfying (2).} \quad (3)$$

In short, for every $c \leq c_{f,x}(e_x)$, there is no perturbation scheme on non-explanatory features that can alter the label of prediction while keeping the perturbing distortion less than c . We call $c_{f,x}(e_x)$ the c -Eval of explanation e_x .

To this point, we have formulated the definition of c -Eval and described our intuition on the connection between c -Eval of an explanation and the importance of the explanatory features. Based on that connection, we propose to use c -Eval as a quantitative metric to evaluate the representative power of neural networks' explanations. Before discussing on computing c -Eval in Section III and strengthening the relationship between c -Eval and the power of explanatory features in Section IV, we now emphasize some properties and several remarks on the usage of c -Eval.

Range of c -Eval. When there is no element in the set of explanatory features, we have $c_{f,x}(e_x) = c_{f,x}(\emptyset)$ is the minimum amount of perturbation onto all input's features to successfully change the original prediction. In this case, the successful perturbation h_{g_f} will return a perturbation known as the *minimally distorted adversarial examples* [19]. On the other hand, when explainer g_f returns all features of the input image, there is no perturbation h_{g_f} can alter the prediction's label and we set $c_{f,x}(x) = \infty$ by convention.

Similar explanations' size. We limit the usage of c -Eval to explanations of the same or comparable sizes. The reason is an explainer can naively include a lot of unnecessary features in its explanations and trivially increase the corresponding c -Eval. However, this restriction does not prevent the usage of c -Eval in evaluating explanations of different explainers. In fact, we can always fix a compactness parameter k (number of input features, number of pixels or number of image's segments as explanatory features) and take the top- k important elements as an explanation. When comparing different explainers using c -Eval, we will specify how compactness parameters of explanations are chosen. For most experiments in this paper, k is chosen to be 10% of the number of input features.

Normalize c -Eval among inputs. Given a compactness parameter k , for different inputs x , the amount of minimum perturbing distortion resulting in successful perturbations can vary significantly depending on the raw features' values and formats. For instance, a RGB image can be encoded using 255 integer values or a range of float values between 0 and 1. Hence, for meaningful statistical results, some experiments use the normalize ratio between c -Eval of e_x and c -Eval of empty explanation, i.e. $C_{f,x}(e_x) = c_{f,x}(e_x)/c_{f,x}(\emptyset)$, to evaluate e_x .

The choice of norm for c -Eval. To our knowledge, there has been no research on which distance metric is optimal to measure the interpretability of explanations. There is also no consensus on the optimal distance metric of human perceptual similarity [19]. Because of the followings reasons, we consider the L_2 -norm, i.e. $p = 2$, throughout this work: (i) L_2 -norm has been used to generate explanation for neural networks'

predictions [3], (ii) our computation of c -Eval is related to the generation of adversarial samples, whose initial work [20] used L_2 -norm, and (iii) there exists efficient algorithms to minimize L_2 -norm in adversarial generation [21], [22].

III. COMPUTING c -EVAL

Given an explanation, it is not straight-forward to compute its c -Eval by using formula (3). Instead, we solve for the successful perturbation scheme with the smallest distortion. Specifically, we compute c -Eval based on the following equivalent definition:

$$c_{f,x}(e_x) = \inf c \quad \text{s.t. } \exists h_{g_f} \text{ satisfying (2).} \quad (4)$$

Based on (4), the c -Eval of explanation e_x can be obtained by solving for the minimum perturbation scheme h_{g_f} on non-explanatory features.

The computation of c -Eval can be summarized through an example shown in Fig. 2. Given an input image and an explanation for the prediction on that image, we compute the minimal perturbing distortion successful perturbation on non-explanatory features of that image using the *Perturbation* block. The c -Eval of the explanation is then approximated by the norm of the difference between the minimal distortion perturbation and the input image.

In Fig. 2, we generate an explanation of LIME explainer for a *Bernese mountain dog* prediction on the given input image. The explanation in this case includes roughly 10% the total number of input pixels. After that, a perturbed instance $h_{g_f}(x)$ is generated using our modified version of Carlini-Wagner (CW) attack [19] where the perturbation is only on the non-explanatory features. Then, the c -Eval is the norm of the difference between the input image and the perturbed image. The reported c -Eval computed in the L_2 -norm is 0.6297. For the sake of demonstration, we construct a "dummy square" of the same size as the LIME explanation, which include the center region of the original image. We consider this mask as an explanation for the prediction and compute the c -Eval for it, which is 0.6154. The c -Eval of LIME is larger than that of the dummy square, i.e. the amount of perturbation required to change the prediction while fixing the explanatory features of LIME is greater. This result aligns with our expectation that LIME explanation should be better than a dummy square in explaining the model's prediction.

For the computation of c -Eval, the only step required further specifications is the Perturbation step (Fig. 2) determining the minimum distortion perturbation on non-explanatory features $h_{g_f}(x)$. We use the CW attack for this step since it has been widely considered as the state-of-the-art algorithm generating minimal distortion adversarial samples of neural networks. In this attack, the solver searches for an optimal difference δ minimizing $\mathcal{D}(x, x + \delta) + \lambda l(x + \delta)$, where $\mathcal{D}$ is a distance metric between the perturbation and the original input, l is a loss function such that $l(x + \delta) \leq 0$ if and only if the label of $x + \delta$ is different from the original label and $\lambda > 0$ is a constant weight used to adjust the priority of the algorithm. Note that δ also need to satisfy the box-constraints, i.e. $x + \delta$

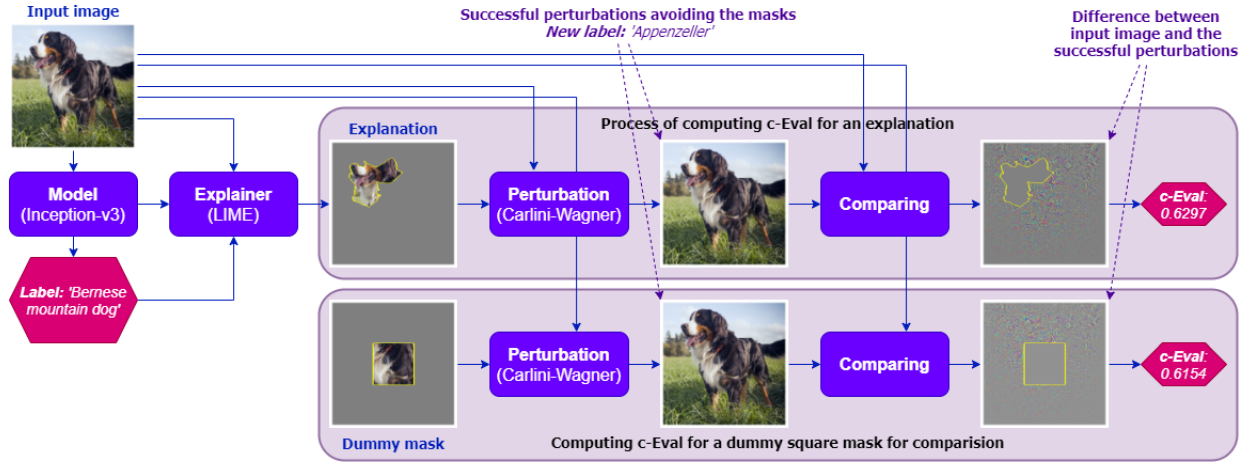


Fig. 2: Example comparing the importance of features including in an explanation and features in a dummy mask using c -Eval.

must be in a valid value-range imposed by the dataset. Then, the optimal perturbation δ is learnt via gradient-descents.

A naive modification of the CW attack so that it only perturbs non-explanatory features is by blocking the backward update on explanatory features in the gradient descents. However, the rate of convergence of a such modification may reduce significantly if many δ_i components with high gradients are blocked. Unfortunately, the situation happens frequently as most existing explainers tend to include high-gradient components as explanatory features.

To overcome this problem, we introduce perturbation variables $\delta_{e_x} \in [0, 1]^{n-|e_x|}$ representing perturbations on non-explanatory features. We then use a mapping $s : [0, 1]^{n-|e_x|} \rightarrow \mathbb{R}^n$ that transforms the perturbation information in δ_{e_x} into δ . The mapping guarantees that for any explanatory feature i , $\delta_i = 0$. By using s , we can guarantee that the optimization steps focus on non-explanatory features. To solve for δ_{e_x} , we use Adam [23] optimizer with the following objective:

$$\mathcal{D}(\mathbf{x}, \mathbf{x} + s(\delta_{e_x})) + \lambda l(\mathbf{x} + s(\delta_{e_x})). \quad (5)$$

One drawback of CW attack is the high running-time complexity. However, from the perspective of c -Eval, we might not need exactly the minimal distortion perturbation to evaluate the explanations. Suppose that we have an algorithm searching for successful perturbations on non-explanatory features of $\mathbf{x}$. If e_x is important to the prediction, it will be difficult for the algorithm to find successful perturbations by perturbing only on $\mathbf{x} \setminus e_x$. Thus, the resulted distortion will be higher than that when e_x is not important. The intuition here is very similar to the definition of c -Eval in previous section. The only difference is in the space of the perturbation schemes. Thus, we extend our definition of c -Eval to the “ c -Eval with respect to a class of perturbing scheme $\mathcal{H}$ ” as follows.

Definition 2. An explainer g_f (or the corresponding explanation e_x) of a prediction $f(\mathbf{x})$ is c -Eval with respect to the class of perturbing schemes $\mathcal{H}$ if no perturbing scheme $h_{g_f} \in \mathcal{H}$ can change the model prediction on $\mathbf{x}$ while keeping the perturbing distortion less than or equal to c .

Definition 2 helps us avoid the difficulty in finding the minimum-distortion perturbation scheme h_{g_f} . Instead of examining all perturbations scheme satisfying the p -norm constraint within distance c , we can focus on the optimal h_{g_f} in a much smaller set of perturbation schemes $\mathcal{H}$. By narrowing down the choices of h_{g_f} , the computation of c -Eval can be tractable without much loss in performance. Specifically, we propose to focus on the set of perturbations generated by the Gradient-Sign-Attack (GSA) [24], and the Iterative-Gradient-Attack (IGA) [25] due to their low running time complexity. Given an image $\mathbf{x}$, GSA sets the perturbation $\mathbf{x}'$ as

$$\mathbf{x}' = \mathbf{x} - \epsilon \cdot \text{sign}(\nabla J_l(\mathbf{x})), \quad (6)$$

where J_l is the l component of the loss function used to train the neural network and ϵ is a small constant. On the other hand, IGA initializes $\mathbf{x}'^{(0)} = \mathbf{x}$ and updates it iteratively as

$$\mathbf{x}'^{(i+1)} = \text{clip}_{\mathbf{x}, \epsilon} \left(\mathbf{x}'^{(i)} - \alpha \cdot \text{sign}(\nabla J_l(\mathbf{x}'^{(i)})) \right) \quad (7)$$

where the clip function ensures that $\mathbf{x}'^{(i)}$ is in the ϵ -neighborhood of the original image. To adopt GSA and IGA into the context of c -Eval where the perturbation is on non-explanatory features, we simply block the backward step of gradient-descent algorithm on explanatory features.

Fig. 3 demonstrates the distortions generated by different attacks. The experiment setup is the same as in the experiment of Fig. 2. The L_2 -norm of the distortions generated by GSA and IGA on LIME explanation are 1.3120 and 0.9804, respectively. The corresponding c -Eval for the dummy square are 1.2962 and 0.9696. We can see that the distortions in GSA and IGA are more spreading out due to the nature of the attacks, which constitutes higher total distortions. Even though the distortions in GSA and IGA are larger than those computed by CW attack, their results still imply that LIME explanation is better than the dummy square and align with our intuition on their explaining power.

Fig. 4 is the scatter plot of c -Eval of 30 explanations in Inception-v3 computed by different perturbations methods.

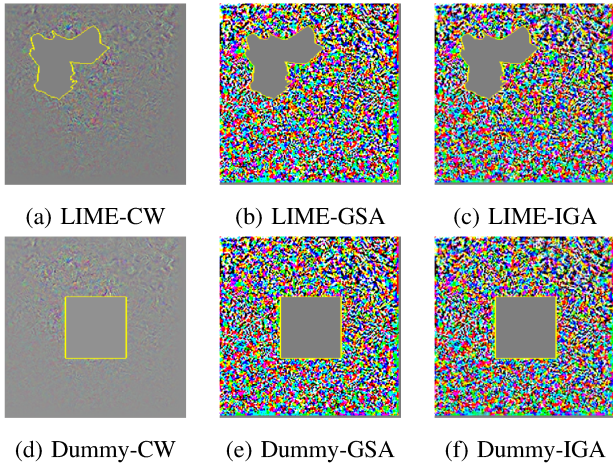


Fig. 3: Distortions between perturbations and the original images. The notations 'LIME' and 'Dummy' stand for LIME explanation and dummy explanation in experiment of Fig. 2.

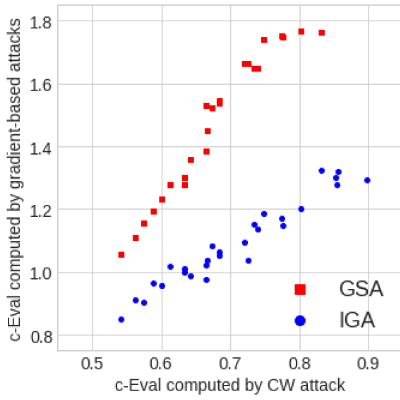


Fig. 4: Scatter plot of c -Eval computed by gradient-based attacks vs c -Eval computed by CW attack.

The results demonstrate the strong correlations of c -Eval computed by GSA as well as IGA to c -Eval obtained from CW attack. Based on these correlations, we will use GSA and IGA instead of CW attack to compute c -Eval for some experiments in this work due to their lower running-time complexity.

IV. C-EVAL AND THE IMPORTANCE OF FEATURES

This section illustrates a relationship between c -Eval and the importance of features returned by local explainers. We first demonstrate this relationship in multi-class affine classifiers. We show that c -Eval determines the minimum distance from the explained data point (the input image) to the nearest decision hyperplane in a lower-dimension space restricted by the choice of explanatory features. A high c -Eval implies that the chosen explanatory features are more aligned with the minimum projection's direction, i.e. they are features determining the prediction on the data point. We further extend the analysis of c -Eval to general non-affine classifiers. Our experiments based on c -Eval suggest an existence of nearly-affine decision surfaces in several well-known classifiers.

A. c -Eval in affine classifiers

We consider an affine classifier $f(\mathbf{x}) = \mathbf{W}^T \mathbf{x} + \mathbf{b}$ where $\mathbf{W}$ and $\mathbf{b}$ are given model's parameters. Given an explanation $e_{\mathbf{x}}$, c -Eval is the solution of the following program:

$$\begin{aligned} \min \quad & \|\delta\|_2 \\ \text{s.t.} \quad & \exists j : \mathbf{w}_j^T (\mathbf{x} + \delta) + b_j \geq \mathbf{w}_{j_0}^T (\mathbf{x} + \delta) + b_{j_0}, \\ & \forall i \in e_{\mathbf{x}}, \delta_i = 0, \end{aligned} \quad (8)$$

where $\mathbf{w}_j$ is the j^{th} column of $\mathbf{W}$, $j_0 = \arg \max_j f(\mathbf{x})$ is the original prediction and δ is the vector of δ_i defined in (1).

When $e_{\mathbf{x}} = \emptyset$, there is no restriction on entries of δ . The optimization program (8) computes the distance between $\mathbf{x}$ and the complement of convex polyhedron P :

$$P = \bigcap_{j=1}^m \{\mathbf{x} : f_{j_0}(\mathbf{x}) \geq f_j(\mathbf{x})\}, \quad (9)$$

where $\mathbf{x}$ is located inside P . The optimal $c_{f,\mathbf{x}}(\emptyset)$ of (8) is a distance from $\mathbf{x}$ to the closest decision hyperplane $\mathcal{F}_j = \{\mathbf{x} : f_{j_0}(\mathbf{x}) = f_j(\mathbf{x})\}$ of P . For the sake of demonstration, Fig. 5 describes an example in 2-dimension space where $c_{f,\mathbf{x}}(\emptyset)$ is plotted in the green line.

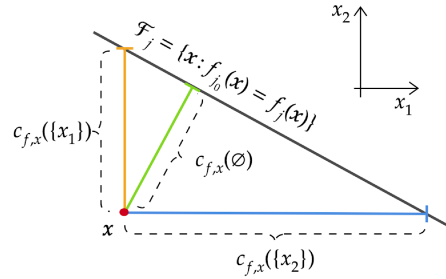


Fig. 5: c -Eval in 2D affine classifier. Explanation $\{x_2\}$ is better than $\{x_1\}$ since the distance from $\mathbf{x}$ to hyperplane $\mathcal{F}_j$ without changing x_2 is larger than that distance without changing x_1 .

For each explanatory feature in $e_{\mathbf{x}}$, the optimization space of (8) is reduced by one dimension. The optimization program (8) then solves for the shortest distance from $\mathbf{x}$ to the complement of polyhedron P in a lower dimension. A more important subset $e_{\mathbf{x}}$ implies more restrictive constraints on the optimization (8) and a larger distortion/distance $\|\delta\|$. In 2-dimension space as depicted in Fig. 5, under an assumption that $\mathcal{F}_j$ is also the closest hyperplane of P to $\mathbf{x}$, $c_{f,\mathbf{x}}(e_{\mathbf{x}} = \{x_1\})$ is the distance from $\mathbf{x}$ to $\mathcal{F}_j$ when the feature x_1 is unchanged. Similarly, the c -Eval $c_{f,\mathbf{x}}(e_{\mathbf{x}} = \{x_2\})$ is the length of the blue line in the figure. In this case, allowing changing x_2 is easier to alter the original prediction j_0 than x_1 , i.e. $c_{f,\mathbf{x}}(\{x_1\}) < c_{f,\mathbf{x}}(\{x_2\})$. It implies that x_2 is more important to the prediction than x_1 .

To this point, we see that under the affine assumption on classifier f , c -Eval of an explanation is the length of the projection from the data point to the decision hyperplanes in the space of non-explanatory features. Therefore, the explanation with high c -Eval contains features whose dimensions are more aligned with the shortest distance vector from the data point to

the decision hyperplane. Thus, c -Eval reflects the importance of features in the explanation.

B. c -Eval in general non-affine classifiers.

For more general classifiers, the set P in equation (9) describing the region of prediction j_0 is no longer a polyhedron. However, our observation based on c -Eval suggests that many well-known image classifiers might be nearly affine in a wide-range of local predictions. Therefore, it is still applicable to evaluate models' explanations using c -Eval.

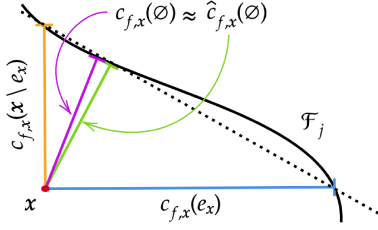


Fig. 6: c -Eval in non-linear classifier. When f is nearly affine, $c_{f,x}(\emptyset) \approx \hat{c}_{f,x}(\emptyset)$.

Our observation is based on a property of c -Eval in affine classifier. Given an explanation e_x , we have shown that $c_{f,x}(e_x)$ is the distance from x to $\mathcal{F}_j$ without changing features in e_x . Similarly, $c_{f,x}(x \setminus e_x)$ is the distance from x to $\mathcal{F}_j$ without changing the complement of e_x . As in the case of 2-dimension in Fig 5, $c_{f,x}(\emptyset)$ is the height to the hypotenuse of the right triangle whose sides are $c_{f,x}(e_x)$ and $c_{f,x}(x \setminus e_x)$. Thus, we have the following equalities:

$$\frac{1}{c_{f,x}(\emptyset)^2} = \frac{1}{c_{f,x}(e_x)^2} + \frac{1}{c_{f,x}(x \setminus e_x)^2} \quad (10)$$

$$\Leftrightarrow c_{f,x}(\emptyset) = \frac{1}{\sqrt{1/c_{f,x}(e_x)^2 + 1/c_{f,x}(x \setminus e_x)^2}}, \quad (11)$$

for any explanation e_x . We denote the expression on the right-hand-side of (11) by $\hat{c}_{f,x}(\emptyset)$.

For non-linear classifiers f , equation (11) does not hold in general. However, if the decision surface $\mathcal{F}_j$ is nearly affine, we should have $c_{f,x}(\emptyset) \approx \hat{c}_{f,x}(\emptyset)$ for all e_x as described in Fig. 6. By testing different classifiers, we observe that this necessary condition hold for many data points of common image classifiers such as Inception-v3 [16], VGG19 [26] and ResNet50 [27]. Specifically, in experiment on Inception-v3 shown in Fig. 7, we generate a 8×8 GCam explanation on the Inception-v3 and iteratively compute $c_{f,x}(e_x)$ and $c_{f,x}(x \setminus e_x)$ using CW attack. Here, we vary the number of explanatory features k in e_x and compute the corresponding $\hat{c}_{f,x}(\emptyset)$ using equation (11). The value of $c_{f,x}(\emptyset)$ is drawn using the purple straight-dot-line for reference. We can see that the two lines for $c_{f,x}(\emptyset)$ and $\hat{c}_{f,x}(\emptyset)$ are close to each other. The experiments on VGG19 and ResNet50 are plotted in Fig. 8. Due to running-time complexity, we use GSA to compute the c -Eval on the same input image as for Inception-v3. Note that the L_2 distortion is computed based on the input

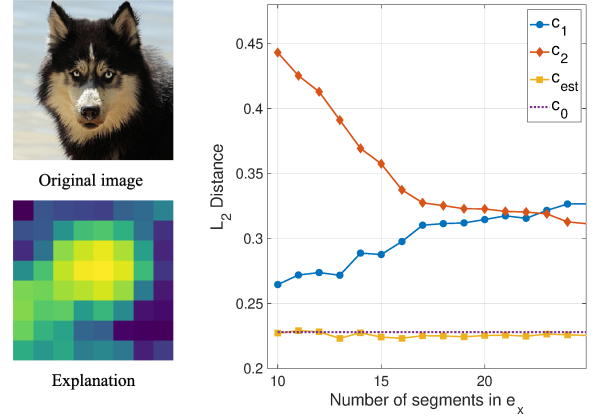


Fig. 7: Example of nearly affine instance on Inception-v3. Here, c_1 , c_2 , c_{est} and c_0 are $c_{f,x}(e_x)$, $c_{f,x}(x \setminus e_x)$, $\hat{c}_{f,x}(\emptyset)$ and $c_{f,x}(\emptyset)$ respectively. Since $c_{f,x}(\emptyset) \approx \hat{c}_{f,x}(\emptyset)$ for all number of segments from the explanation, we might infer that the decision surface is nearly affine in this example.

space of each model. We can see that $\hat{c}_{f,x}(\emptyset)$ and $c_{f,x}(\emptyset)$ are close to each others in both models. It is interesting that different models share this same property, which encourage us to use c -Eval to evaluate explanations of those classifiers.

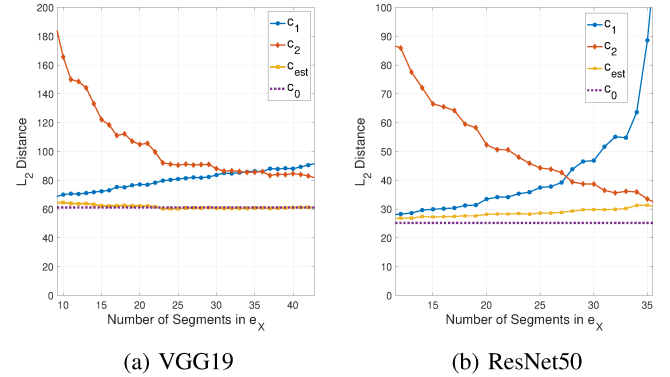


Fig. 8: The condition $c_{f,x}(\emptyset) \approx \hat{c}_{f,x}(\emptyset)$ also holds for VGG19 and ResNet50, which also suggests the existence of nearly-affine decision surfaces in these models.

V. BEYOND c -EVAL

In the following Subsection V-A, we introduce the c -Eval plot, which is a visualization of explainers' behavior on a given input based on c -Eval. Using examples on LIME explainer, we demonstrate that c -Eval plot helps us determine appropriate tuning parameters for the explainer and strengthens the usage of c -Eval in evaluating the importance of explanatory features. On the other hand, since c -Eval relies on the generation of successful perturbations, Subsection V-B discusses c -Eval's behaviors in adversarial-robust models. We show that c -Eval computed in adversarial-robust models is strongly correlated with its non-robust counterpart. The result implies that c -Eval is applicable in adversarial-robust models.

A. *c*-Eval plot

In Section II, we restrict the *c*-Eval analysis on explanations of similar sizes. That restriction is just for a fair comparison among explanations of different explainers. In practice, the explanation's size is normally determined based on the applications and the specific inputs. Given an explainer, by varying explanation's size k , we obtain a sequence of explanations and their corresponding *c*-Eval. Therefore, on a given input image, each explainer will be associated with a sequence of *c*-Eval values. By plotting this sequence as a function of k , we can observe the behaviors of explainers on that input and select an appropriate size for the explanation accordingly. We call the resulting plot the *c*-Eval plot.

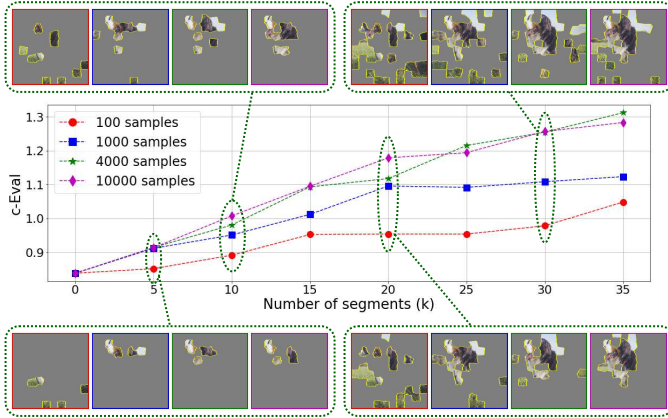


Fig. 9: *c*-Eval plot of LIME with different sample rates. Higher sample rates result in better explanations and *c*-Eval plot reflects that expectation.

We now discuss an experiment based on the *c*-Eval plot to heuristically show that it correctly evaluates the explanations and improve our understanding on the behavior of the explainer. Specifically, we study the LIME explainer with different number of samplings. In LIME, the sampling size determines how many perturbations are conducted in finding the explanation. The higher the number, the better the explanation and the higher the running time complexity [3]. Since there is no concrete rule on how this parameter should be chosen, how can we verify that a LIME explanation is free from under-sampling error? What is an appropriate number of explanatory features to explain the prediction? In the following, we show how *c*-Eval plot reflects the impact of the number of samplings on LIME's performance and help us determine appropriate values for that parameter.

The experiments are conducted on Inception-v3 with the input image as in the experiment in Fig. 2. We first segmented that image into 100 feature segments. Then, we explain them using LIME with 100, 1000, 4000 and 10000 samples. We plot the *c*-Eval sequences $\{c_{f,x}(e_{\mathbf{x}}^k)\}_{k=1}^n$, i.e. the *c*-Eval plot in fig. 9. For some numbers of segments (the explanation's size $k = 5, 10, 20$ or 30) and for each setting of LIME (red for 100, blue for 1000, green for 4000 and purple for 10000 samples), we provide the corresponding

explanations for illustration in Fig. 9. The result shows distinct gaps in the *c*-Eval among different numbers of samplings. It is clear that the higher the number of samples, the higher the *c*-Eval. This observation is consistent with our expectation that the explanations at higher numbers of samples are better. Additionally, using *c*-Eval plot, we can deduce that there is not much improvement in the explanations' quality by increasing the number of samples from 4000 to 10000. This implies *c*-Eval can be used as a metric to support automatically tuning of explainer's parameters. It also helps us gain trust in LIME in the sense that, if we aim for top-5 important features among 100 features, LIME with 2000 samples might be reliable since there is not much gain in *c*-Eval by increasing that number from 1000 to 10000. We also observe that, given the number of samples, there is a diminishing return in *c*-Eval after a certain number of explanatory features. For instance, if we only use 1000 samples, 20 explanatory features (roughly 20% of the input image) is enough to explain the prediction.

B. *c*-Eval on adversarial-trained models

Since *c*-Eval is computed based on adversarial generation, there might exist concerns regarding the applications of *c*-Eval on adversarial-robust models. First, as adversarial-robust models are more resistant to perturbations, is it feasible to generate successful perturbations on robust models? Second, if we are able to obtain those perturbations, are the *c*-Eval values of the corresponding explanations reliable? In the following, we address those concerns through experiments on MNIST dataset using the LeNet model [28]. Specifically, we show that the *c*-Eval computed on non-robust and robust models have strong correlation. This correlation implies that the behaviors of *c*-Eval are similar on non-robust and robust models.

We use Advtorch [22], a Python toolbox for adversarial robustness research, to train three LeNet classifiers on the MNIST dataset. The first model, denoted as non-robust model, is trained normally. The second is alternatively trained between images from MNIST and the corresponding adversarial samples generated at each iteration. Here, the normalized L_2 -norm distortion between each adversarial sample and its original image is bounded by $\epsilon = 0.3$. The third model is trained in the same manner as the second but the distortion bound ϵ is set to 0.5. All three classifiers archive more than 95% accuracy on test set. For the two adversarial-trained models, their accuracy on adversarial samples are all greater than 94%.

Using 4000 images in the test set, we generate their predictions made by the three LeNet classifiers and the corresponding top-10% LIME explanations. For all three classifiers, we are able to obtain the successful perturbations using IGA and the corresponding *c*-Eval for all explanations. The successful perturbations for all inputs of adversarial-trained models can be computed because the models are only robust against adversarial with bounded distortion. In *c*-Eval, the perturbations are not limited by the amount of distortion.

Fig. 10 is the scatter plot of *c*-Eval of 300 (randomly chosen from 4000) explanations of predictions from the three classifiers. The Pearson correlations between *c*-Eval of the first

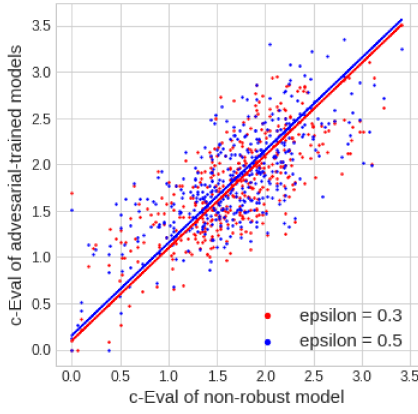


Fig. 10: Correlation between c -Eval on non-robust and adversarial-trained models.

model and those of the other two adversarial-trained models are 0.765 and 0.764 respectively. We assert this is a fairly high correlation when we take into account that these are three models trained separately. In fact, on average, less than 75% of the explanatory features are shared between non-robust model and any of the others. We also observe the model with higher bound in the distortion in the training has higher average c -Eval. This aligns with our intuition on how c -Eval is computed.

VI. EXPERIMENTAL RESULTS

In this section, we use c -Eval to experimentally evaluate explanations generated by different feature-based local explainers on small gray-scale hand-writing images MNIST [18] and large color object images Caltech101 [29]. We also provide experimental results showing that evaluations of explanations based on c -Eval on MNIST dataset align with previous results obtained from log-odd scoring function [5], which is specifically designed for MNIST dataset only. To demonstrate the statistic behavior of c -Eval on large number of samples, the reported c -Eval is not precisely $c_{f,x}(e_x)$ but the ratio of $c_{f,x}(e_x)$ over $c_{f,x}(\emptyset)$. This ratio is indicated by the notation $C(e_x)/C_0$ in the legend of each figure. The ground-truth quality rankings of explanations are obtained from previous results in assessing explainer's performance using human-based experiments [2], [5]. The studied classifier models and explainers are selected based on those previous experiments accordingly. The system specifications and the codes for our implementations are specified in subsection VI-A. For reference, we also provide experiments on small-size color image dataset CIFAR10 [30], which can be found in our other report [31]. In the Appendix of that report, we also provide some explanations of images from MNIST, Caltech101 and CIFAR10 datasets along with their computed c -Eval for visualization.

A. System specifications and source code

Our experiments in this paper are conducted in Python. The computing platform is a Linux server equipped with two Intel Xeon E5-2697 processors supporting 72 threads. Our system memory comprises twelve 32 GB DDR4 sticks, each operates

at 2400 MHz. The source code for our experiments can be found at [32].

B. Simulations on MNIST dataset

For the MNIST dataset [18], we study 8 different feature-based local explainers: LIME [3], SHAP [2], GCam [4], DeepLIFT (DEEP) [5], Integrated Gradients [33], Layerwise Relevance Propagation (LRP) [6], Guided-Backpropagation (GB) [7] and Simonyan-Gradient (Grad) [8]. Followings are brief descriptions of these explainers.

In LIME, the importance of each picture segment is approximated with a heuristic linear function using random perturbation. SHAP, which relies on the theoretical analysis of Shapley value in game theory, assigns each pixel a score indicating the importance of that pixel to the classifier's output. Since SHAP is a generalized version of LIME, we expect SHAP explanation to be more consistent with the classifier than LIME, hence SHAP's c -Evals are expected to be higher statistically. Previous work [2] also provided human-based experiments to support this claim. DeepLIFT, Integrated Gradients, LRP, GB and Grad are backward-propagation methods to evaluate the importance of each input neuron to the final output neurons of the examined classifier. Previous experiment results using log-odds function in [5] suggest that GB and Grad perform worse than the other three in MNIST dataset. The final studied explainer GCam is an image explainer designed specifically for fully-connected convolutional networks. It exploits the last convolution layer to explain the model's prediction. Since GCam is not designed for classifiers of low-resolution images, we expect its performance and the corresponding c -Eval in the MNIST dataset are limited.

Our experiments on the MNIST dataset are conducted in pixel-wise manner, i.e. the outputs of explainers are image pixels. For each input image, each explainer except LIME is set to return 10% the number of image pixels as explanation. For LIME, since the algorithm always returns image segments as explanations, we set the returned pixels to be as close to 10% of the total number of pixels as possible. On another note, the implementations of LRP are simplified into Gradient $\times$ Input based on the discussion in [5]. The c -Eval and the statistical results of explanations are reported in Fig. 11.

Different classifiers: Figs. 11a and 11b are the distributions of c -Evals on 1000 explanations in MNIST dataset on classifier 1 provided by [2] and classifier 2 provided by [5]. The notation I5 and I10 indicate the Integrated-Gradient method with 5 and 10 interpolations [33]. We can see that the evaluation based on c -Eval is consistent between classifiers as well as previous attempts of evaluating explainers in [2] and [5]. For the consistency in the behavior of c -Eval and log-odds function in [5], please see the discussion in subsection VI-D.

Different gradient-based perturbation schemes: Figs. 11c and 11d demonstrate the usage of IGA instead of GSA as shown in Figs. 11a and 11b. Comparing the distributions in Fig. 11c to Fig. 11a and Fig. 11d to Fig. 11b, the relative c -Eval of explainers are similar between perturbation schemes and consistent with previous experiments in Fig. 4. Thus,

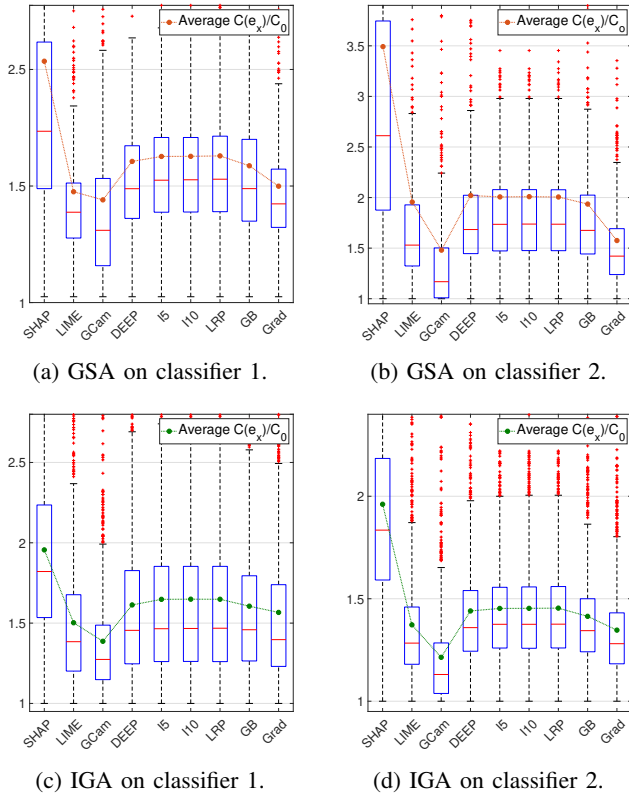


Fig. 11: The distributions and the averages of c -Eval of 8 explainers on classifier 1 provided by [2] and on classifier 2 provided by [5] on 1000 images in MNIST dataset.

the computed c -Evals using IGA also reflect the explainers' performance. Finding optimal perturbation schemes resulting in a good measurement of c -Eval is not considered in this work; however, the experiments suggest that non-optimal perturbation schemes can be used to obtain reasonable measurement of c -Eval.

C. Simulations on Caltech101 dataset

For experiments on large images, we study the performance of LIME, SHAP, GCam, DeepLIFT on 700 images in Caltech101 dataset [29] with the VGG19 classifier [26]. As LIME, SHAP, and GCam explainers are designed for medium-size to large-size images, we expect they should outperform DeepLIFT. Furthermore, the results from [2] implies SHAP should perform better than LIME. On the other hand, as GCam are designed for fully-connected convolution networks (e.g. VGG19), we expect its performance here to be much better than that in previous experiments on MNIST dataset.

Segment-wise features are used on the Caltech101 dataset. Since the outputs of some explainers are importance weights of pixels, we convert them all into a subsets of image segments as explanations for fair comparison. Specifically, the importance weight of each segment is the sum of the importance weights of all pixels inside that segment. Then, the top k segments with maximum weight are selected as the segment-wise explanation

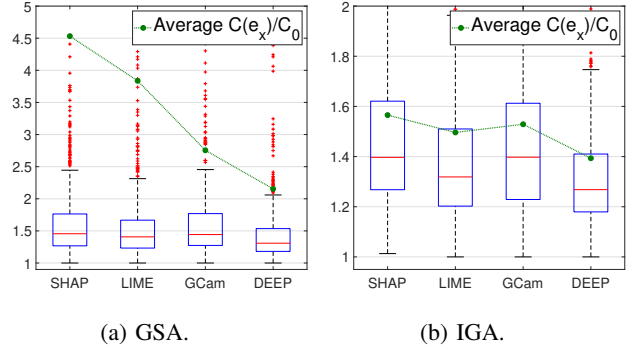


Fig. 12: Distributions of c -Eval computed by GSA and IGA for four explainers in Caltech101 dataset.

of the examined explainer. For the results in Fig. 12, the number of segments is selected such that roughly 20% of the original input image is covered by the explanation.

The computed c -Eval in our experiments on Caltech101 are reported in Fig. 12a and Fig. 12b. Here, we use GSA and IGA to compute c -Eval respectively. The observation is that the statistical behavior of c -Eval aligns with our expectation on the performance of all four explanation method on this dataset. In fact, by design, GCam is expected to perform better in Caltech101 than MNIST. On the other hand, DeepLIFT is not designed and stress-tested on larger models trained on dataset that are significantly bigger than MNIST. More will be discussed in Subsection VI-E. For the improvement of GCam and the degradation of DeepLIFT from the MNIST dataset and CIFAR10 to the Caltech101 dataset, For better intuition on the improvement of GCam and the degradation of DeepLIFT from the MNIST dataset and CIFAR10 to the Caltech101 dataset, we suggest readers check the our examples shown in the Appendix of our other report [31].

D. Similarity of c -Eval and log-odds functions in MNIST

To evaluate importance scores obtained by different methods on MNIST dataset, the authors of DeepLIFT designs the log-odds function as follows. Given an image that originally belongs to a class, they identify which pixels to erase to convert the original image to other target class and evaluate the change in the log-odds score between the two classes. The work conducted experiments of converting 8 to 3, 8 to 6, 9 to 1 and 4 to 1. In Fig. 13, we adopt c -Eval into the MNIST dataset to compare c -Eval of explainers with the corresponding log-odds scores. The figure displays the c -Eval of studied explainers on images with predictions 4, 8 and 9 respectively. We conduct the experiments using both GSA and IGA perturbation schemes. Besides the DeepLIFT in experiments for label 4 and 8, all relative ranking of explainers in c -Eval is consistent with the ranking resulted from log-odds computations shown in [5]. This result implies that our general frameworks of evaluating explainers based on c -Eval are applicable to this specific study on the MNIST dataset.

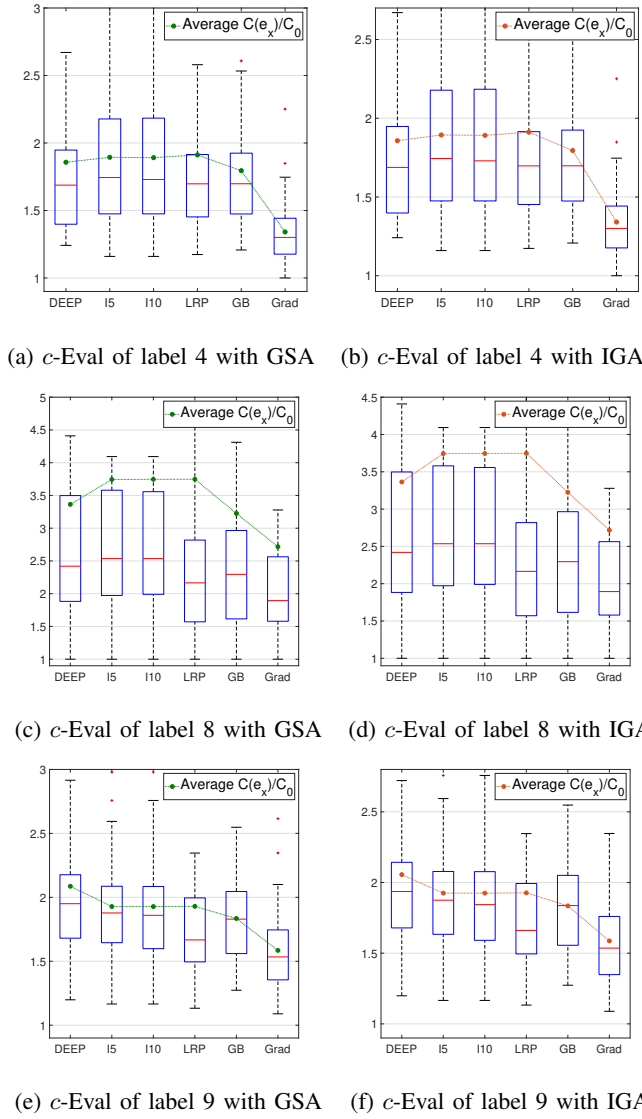


Fig. 13: We compute the c -Eval for 6 explainers on 1000 images of MNIST for labels 4, 8 and 9 to show the similarity between c -Eval and log-odds function in [5].

E. Overall evaluations of explanations using c -Eval

Many interesting results and deductions can be drawn from experiments on MNIST and Caltech101. The result is also consistent with our experiments on CIFAR10 reported in [31]. We discuss several key observations in the followings.

Our first comment is about the correlation of c -Eval and the portion of predicted object captured by different explanations. In CIFAR10 and especially Caltech101 (see [31]), it is clear to us that most explanations containing the essential features of the predicted label have high c -Eval.

Our second attention is on the relative performance of GCam in three datasets. Since GCam is designed for convolutional neural networks such as the VGG19, we expect high relevant explanations from GCam in its experiments on

Caltech101. However, as GCam exploits the last layer of the neural networks to generate the explanations [4], we have low expectation on its capability of explaining predictions on MNIST and CIFAR10 dataset. The reason is that the models used in those two later dataset are too different from the VGG19. In fact, the adaptation of VGG19 on CIFAR10 [34] contains only 4 neurons in the last convolutional layer, which results in only 4 regions of the images that GCam can choose as region of high important. The distributions of c -Eval of GCam shown in Fig. 12b also reflect that expectation.

DeepLIFT is a back-propagation method and it is not only sensitive to the classifier structure but also the selection of reference image [5]. The experimental setups of DeepLIFT in the MNIST dataset shown in Fig. 11 are taken directly from the source code of the explainer's paper. Our adoptions of DeepLIFT to CIFAR10 and Caltech101 are conducted without calibration on the reference image as the calibration procedure for color images is not provided. This might be the reason for the degradation of explainer's quality in these two datasets. It is clear that c -Eval captures this behavior.

Our final remark is on the exceptionally high c -Eval of SHAP shown in all three datasets. This result encourages us to take a deeper look at explanations produced by SHAP. From our observation, it shows that SHAP captures some important features that are overlooked by others. Let's consider the explanation of number 4 as an example. SHAP is the only explainer detecting that the black area on top of number 4 is important (see examples in [31]). In fact, this area is essential to the prediction since, if these pixels are white instead of black, the original prediction should be 9 instead of 4. Without the c -Eval computations, it is non-trivial to identify this beneficial behavior in SHAP explainer.

VII. CONCLUSIONS AND FUTURE WORKS

In this paper, we introduce c -Eval to evaluate explanations of various feature-based explainers. Extensive experiments show that c -Eval of explanation reflects the importance of features included in the explanation. This study leads to several interesting research questions for the future work. For example, the distributions of c -Eval in Fig. 11 advocates that there is a fundamental difference between the quality of black-box explainers (SHAP, LIME and GCam) and back-propagation explainers (DEEP, Integrated Gradients, LRP, GB and Grad), which is ambiguous prior to this work. From the novelty of c -Eval, we expect that knowledge on the explanation maximizing c -Eval will offer us a much clearer view on predictions made by modern neural networks.

ACKNOWLEDGMENTS AND DISCLOSURE OF FUNDING

This work was supported in part by the National Science Foundation Program on Fairness in AI in collaboration with Amazon under award No. 1939725.

REFERENCES

- [1] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, p. 115, Jan 2017. [Online]. Available: <https://doi.org/10.1038/nature21056>
- [2] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774. [Online]. Available: <http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf>
- [3] 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*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1135–1144. [Online]. Available: <https://doi.org/10.1145/2939672.2939778>
- [4] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 618–626.
- [5] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017, pp. 3145–3153. [Online]. Available: <http://proceedings.mlr.press/v70/shrikumar17a.html>
- [6] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLOS ONE*, vol. 10, no. 7, pp. 1–46, 07 2015. [Online]. Available: <https://doi.org/10.1371/journal.pone.0130140>
- [7] J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," in *ICLR (workshop track)*, 2015. [Online]. Available: <http://lmb.informatik.uni-freiburg.de/Publications/2015/DB15a>
- [8] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *Workshop at International Conference on Learning Representations*, 2014.
- [9] D. Smilkov, N. Thorat, B. Kim, F. B. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," *Workshop on Visualization for Deep Learning, ICML*, vol. abs/1706.03825, 2017. [Online]. Available: <http://arxiv.org/abs/1706.03825>
- [10] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 2017, pp. 3319–3328. [Online]. Available: <http://proceedings.mlr.press/v70/sundararajan17a.html>
- [11] M. Robnik-Šikonja and I. Kononenko, "Explaining classifications for individual instances," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 5, pp. 589–600, May 2008.
- [12] E. Štrumbelj, I. Kononenko, and M. Robnik Šikonja, "Explaining instance classifications with interactions of subsets of feature values," *Data Knowl. Eng.*, vol. 68, no. 10, pp. 886–904, Oct. 2009. [Online]. Available: <http://dx.doi.org/10.1016/j.datak.2009.01.004>
- [13] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *AAAI*, 2018.
- [14] Z. C. Lipton, "The mythos of model interpretability," *Queue*, vol. 16, no. 3, p. 31–57, Jun. 2018. [Online]. Available: <https://doi.org/10.1145/3236386.3241340>
- [15] B. Kim, E. Glassman, B. Johnson, and J. Shah, "iBCM : Interactive Bayesian case model empowering humans via intuitive interaction," 2015.
- [16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2818–2826.
- [17] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K. Müller, "Evaluating the visualization of what a deep neural network has learned," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 11, pp. 2660–2673, Nov 2017.
- [18] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010. [Online]. Available: <http://yann.lecun.com/exdb/mnist/>
- [19] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, May 2017, pp. 39–57.
- [20] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014. [Online]. Available: <http://arxiv.org/abs/1312.6199>
- [21] J. Rauber, W. Brendel, and M. Bethge, "Foolbox v0.8.0: A Python toolbox to benchmark the robustness of machine learning models," *CoRR*, vol. abs/1707.04131, 2017. [Online]. Available: <http://arxiv.org/abs/1707.04131>
- [22] G. W. Ding, L. Wang, and X. Jin, "AdverTorch v0.1: An adversarial robustness toolbox based on pytorch," *arXiv preprint arXiv:1902.07623*, 2019.
- [23] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.
- [24] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015. [Online]. Available: <http://arxiv.org/abs/1412.6572>
- [25] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *ICLR Workshop*, 2017. [Online]. Available: <https://arxiv.org/abs/1607.02533>
- [26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.
- [27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [28] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [29] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *2004 Conference on Computer Vision and Pattern Recognition Workshop*, pp. 178–178, 2004.
- [30] A. Krizhevsky, "Learning multiple layers of features from tiny images," *University of Toronto*, 05 2012.
- [31] M. N. Vu, T. D. Nguyen, N. Phan, R. Gera, and M. T. Thai, "c-eval: A unified metric to evaluate feature-based explanations via perturbation," 2020. [Online]. Available: <https://arxiv.org/abs/1906.02032>
- [32] [Online]. Available: <https://drive.google.com/drive/folders/1VtexIqMnxUawmP5857EOtFhpt64RyqTY?fbclid=IwAR1Rr7WyKtlqHkXHqFbG5ZByNmJy4bAtHHsAn4vQYNwbUcr6VXSsBfJ2iU>
- [33] M. Sundararajan, A. Taly, and Q. Yan, "Gradients of counterfactuals," *CoRR*, vol. abs/1611.02639, 2016. [Online]. Available: <http://arxiv.org/abs/1611.02639>
- [34] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pp. 730–734, 2015.