

---

# Evaluating Explainers via Perturbation

---

**Minh N. Vu**  
University of Florida  
Gainesville, Florida, USA  
minhvu@ufl.edu

**Truc D. Nguyen**  
University of Florida  
Gainesville, Florida, USA  
truc.nguyen@ufl.edu

**NhatHai Phan**  
NJIT  
Newark, New Jersey, USA  
phan@njit.edu

**Raluca Gera**  
Naval Postgraduate School  
Monterey, California, USA  
rgera@nps.edu

**My T. Thai\***  
University of Florida  
Gainesville, Florida, USA  
mythai@ufl.edu

## Abstract

Due to high complexity of many modern machine learning models such as deep convolutional networks, understanding the cause of model’s prediction is critical. Many explainers have been designed to give us more insights on the decision of complex classifiers. However, there is no common ground on evaluating the quality of different classification methods. Motivated by the needs for comprehensive evaluation, we introduce the *c*-Eval metric and the corresponding framework to quantify the explainer’s quality on feature-based explainers of machine learning image classifiers. Given a prediction and the corresponding explanation on that prediction, *c*-Eval is the minimum-power perturbation that successfully alters the prediction while keeping the explanation’s features unchanged. We also provide theoretical analysis linking the proposed parameter with the portion of predicted object covered by the explanation. Using a heuristic approach, we introduce the *c*-Eval plot, which not only displays a strong connection between *c*-Eval and explainers’ quality, but also serves as a low-complexity approach of assessing explainers. We finally conduct extensive experiments of explainers on three different datasets in order to support the adoption of *c*-Eval in evaluating explainers’ performance.

## 1 Introduction

With the pervasiveness of machine learning in many emerging domains, especially in critical applications such as healthcare or autonomous systems, it is utmost important to understand why a machine learning model makes such a prediction. For example, deep convolutional neural networks have been able to classify skin cancer at level of competence comparable to dermatologists [1]. However, doctors cannot act upon 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, several machine learning explainers that support users in interpreting the predictions of complex models have been studied, such as SHAP [2], LIME [3], Grad-CAM [4], and DeepLIFT [5], among others [6–13].

Despite this very recent development of machine learning explainers, none of them has a theoretical guarantee on the explanation’s quality. These explainers have been only evaluated through a small set of human-based experiments which apparently does not imply the global guarantee on explainer’s performance [2, 5]. Another effort of evaluating explainers, Shrikumar *et. al* uses the log-odds score to measure the difference between the original instance and the modified image whose vital pixels

---

\*Corresponding author

are erased [5]. The log-odds method is only applicable to small gray-scale images like MNIST [14] and there is no theoretical guarantee or rigorous analysis for this method. Based on this lack of comprehensive studies on the quality of explanation, there is a clear need for standard tools and methods to evaluate machine learning model explainers [15].

However, theoretically evaluating explainers remains as a daunting task [15, 16] due to the following reasons. First of all, the model explainers can be very diverse in general. Figure 1 shows an example of three explainers for the prediction “Pembroke” provided by Inception-v3 classifier [17]. All of them highlight the region containing the Pembroke; however, their presentation vary from picture segments in LIME [3], heat-map in Grad-CAM (GCam) [4] to pixel importance-weights in SHAP [2]. In addition to the variety in presentation, explainers might be designed for different objectives. There is a fundamental trade-off between the understanding and the accuracy of the represented features [2, 3]. One explanation can be utilized for end-users to interpret, but a certain degree of consistency with the original model might be lost. The diversity in presentation and objectives constitutes a great challenge in evaluating different explainers.

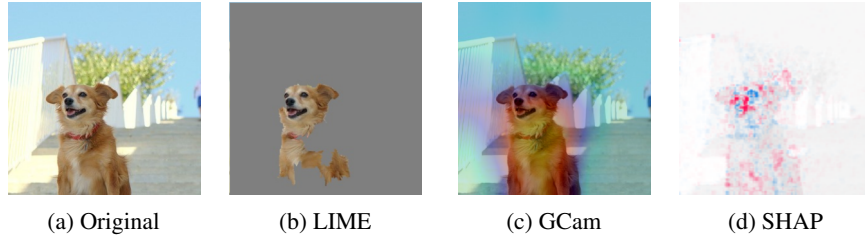


Figure 1: Different feature-based explanations of the prediction "Pembroke".

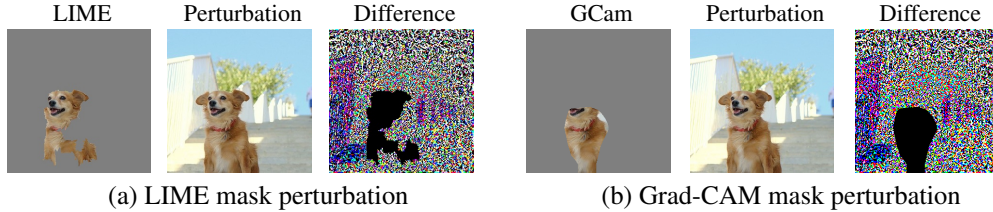


Figure 2: Perturbations are generated by perturbing pixels outside of LIME/GCam explanation region. The power of the Difference images are upper bounds of  $c$ -Eval.

In this research, we focus on evaluating local explainers which are used to interpret individual predictions of black box machine learning models. We introduce a novel metric,  $c$ -Eval, to evaluate the quality of feature-based local explanation. We exploit the intuition that a feature-based local explanation has high quality only if it is difficult to change the prediction when the explanation features are kept intact. The quality of an explanation is quantified by the minimum amount of perturbation on features that are not in the explanation region in order to alter the prediction. We further provide theoretical derivations showing that the portion of the predicted object which must be captured by an explanation is an increasing function of  $c$ -Eval. A low-complexity approach of evaluating explainers based on  $c$ -Eval and experiments are also conducted extensively for different explainers.

We demonstrate the above concepts of  $c$ -Eval via a toy example as shown in Fig. 2. We first cut out 10% of the most important segments returned by two explainers of the Pembroke image. After that, we generate the perturbed instances while keeping the explanation region unchanged. These non-perturbing regions are the black areas in the Difference images. The  $c$ -Eval is computed by using the  $L_2$ -norm distance between the perturbed instance and the original picture. In this example, the  $c$ -Eval of LIME is 1.25 times larger than that of GCam, i.e. the amount of perturbation required to change the prediction while fixing the explanation region of LIME is greater. We claim that LIME includes features explaining this prediction better than GCam. By simply observing Fig. 2, it is clear that LIME explanation is better in this case.

Our contributions in this research can be summarized as follows:

1. We introduce  $c$ -Eval and the corresponding frameworks to evaluate the quality of any feature-based image explainer.

2. We provide the theoretical object coverage guarantee based on  $c$ -Eval: higher  $c$ -Eval implies a larger portion of the predicted object is guaranteed to be included in the explanation.
3. We develop a low-complexity approach using  $c$ -Eval plot to evaluate and visualize explainer’s quality. Assuming the existence of an unknown ideal scoring function that gives better explainers higher scores, we provide heuristic experiments showing that the  $c$ -Eval plot possesses many desirable properties of that ideal scoring function.
4. We conduct extensive experiments on  $c$ -Eval on 8 explainers, using different classifier models and perturbation schemes to validate the usage of  $c$ -Eval in evaluating explainers.

The rest of the paper is organized as follows. Section 2 introduces the notations and formulates  $c$ -Eval, the unified metric to evaluate feature-based explainers. In Section 3, we establish the object coverage guarantee of explainers based on the computed  $c$ -Eval. Due to an expensive computation of the object coverage guarantee, we propose the  $c$ -Eval plot, a low-complexity approach to evaluate explainers in Section 4. Section 5 provides numerical results to support the validity of  $c$ -Eval. Finally, Section 6 concludes the paper with a discussion on future directions.

## 2 Unified evaluation of explainers: $c$ -Eval

In this section we provide the detailed formulation of  $c$ -Eval. We first specify the classifiers, the explainers, and the perturbation schemes.

Given a classifier  $f$ , we denote the set of input features by  $\mathbf{x} = \{x_1, \dots, x_n\} \in \mathcal{X}$  and the corresponding predicted label by  $f(\mathbf{x}) = \mathbf{y} = \{y_1, \dots, y_m\}$ . The final output label is  $\arg \max_{1 \leq j \leq m} y_j$ . A local explanation of classifier  $f$  is  $g_f(\mathbf{x}) = e_{\mathbf{x}} \subset \mathbf{x}$ , which is the set of features explaining the prediction  $\mathbf{y} = f(\mathbf{x})$ . The function  $g_f$  is called the explainer. We denote a perturbation scheme  $h : \mathcal{X} \rightarrow \mathcal{X}$  that perturbs all features of  $\mathbf{x}$  outside of the explanation  $e_{\mathbf{x}}$  into  $\bar{\mathbf{x}}$  as follows

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

A successful perturbation scheme  $h$  on explainer  $g_f$  of instance  $\mathbf{x}$  under power constraint  $c$  is a perturbation scheme such that

$$\arg \max_{1 \leq j \leq m} f(h(\mathbf{x})) \neq \arg \max_{1 \leq j \leq m} f(\mathbf{x}) \quad \text{and} \quad \sum_{i=1}^n \delta_i^2 \leq c. \quad (2)$$

Condition (2) implies that the perturbation satisfies power constraint  $c$  and alters the original prediction of the model on instance  $\mathbf{x}$ . At this point, we are ready to state the definition of  $c$ -Eval.

**Definition 1.** An explainer  $g_f$  (or explanation  $e_{\mathbf{x}}$ ) on instance  $\mathbf{x}$  of classifier  $f$  is  $c$ -Eval at  $\mathbf{x}$  if no perturbing scheme  $h$  satisfying condition (2) successfully changes the original label of instance  $\mathbf{x}$ . We say that an explainer  $g_f$  (or explanation  $e_{\mathbf{x}}$ ) is  $c$ -Eval if  $g_f$  (or  $e_{\mathbf{x}}$ ) is  $c$ -Eval at all  $\mathbf{x}$ .

Intuitively, a good explainer is an explainer with high  $c$ -Eval. For simplicity, we denote  $c_{f,\mathbf{x}}(e_{\mathbf{x}})$  the maximum power  $c$  such that  $e_{\mathbf{x}} = g_f(\mathbf{x})$  is  $c$ -Eval at  $\mathbf{x}$ . It is noteworthy to mention that when the explainer  $g_f$  returns nothing,  $c_{f,\mathbf{x}}(e_{\mathbf{x}}) = c_{f,\mathbf{x}}(\emptyset)$  is the minimum power to perturb the picture to successfully change the original prediction. When  $g_f$  outputs all pixels of the picture, there is no way to successfully perturb the picture, i.e.  $c_{f,\mathbf{x}}(\mathbf{x}) = \infty$ .

Unfortunately, finding the optimal (minimum power) perturbing scheme  $h$  for all instances is intractable. To cope with this issue, given a class of perturbing schemes  $\mathcal{H}$ , we extend our definition of  $c$ -Eval to the “ $c$ -Eval with respect to  $\mathcal{H}$ ” as follows.

**Definition 2.** An explainer  $g_f$  (or explanation  $e_{\mathbf{x}}$ ) on instance  $\mathbf{x}$  of classifier  $f$  is  $c$ -Eval with respect to class of perturbing scheme  $\mathcal{H}$  if there is no perturbing scheme  $h \in \mathcal{H}$  satisfying condition (2) that successfully changes the label of instance  $\mathbf{x}$ .

**Perturbation schemes.** Throughout this work, the considered classes of perturbations are the conventional Gradient-Sign-Attack (GSA) [18] and the Iterative-Gradient-Attack (IGA) implemented by Foolbox [19]. We adopt these perturbation schemes because of their direct connection with the  $L_2$ -norm constraint (2). Without stated otherwise, the  $c$ -Eval computation in this work is carried out by the Gradient-Sign-Attack due to its low complexity.

### 3 Object coverage guarantee of explainers

To address the quality of feature-based local explainers in image classifiers, we introduce the so-called object coverage of explainers. The intuition is that when a classifier predicts a label for an image of an object, one of the most essential and fundamental elements constituting that prediction is the existence of that object in the picture. Consequently, a good explanation for that prediction must capture the object or some key features of that object. This observation suggests there is a strong correlation between the quality of an explanation and the portion of the predicted object captured by the explanation. In the remaining of this section, we will formalize this intuition, establish a connection between  $c$ -Eval and the object coverage of an explainer, and finally obtain the key Theorem 2 on the object guarantee of an explainer.

Given picture  $x$  of single object  $o$  on a background  $b$  where  $o \cap b = \emptyset$ , the coverage of feature-based explanation  $e_x = g(x)$  on the correct prediction  $\arg \max_{1 \leq j \leq m} y_j = j_o$  is the portion of the area of the object captured in the explanation:  $\text{cov}(e_x, o) = |e_x \cap o|/|o|$ . Since we only consider pictures with single object  $o$  only, we use  $\text{cov}(e_x)$  instead of  $\text{cov}(e_x, o)$  for simplicity. As the prediction of  $f(x)$  is object  $o$ , we claim that explanation  $e_x^{(1)}$  is better than  $e_x^{(2)}$  if  $\text{cov}(e_x^{(1)}) > \text{cov}(e_x^{(2)})$  and  $|e_x^{(1)}| \leq |e_x^{(2)}| \leq r|x|$ , where  $r$  defines how much of the original picture is returned as the explanation.

Based on the object coverage, in the next steps, we will establish the connection between  $c$ -Eval and the object coverage of explainer, which implies the interdependence between  $c$ -Eval and explainers' quality. To obtain such guarantee, we first bound the minimum power to generate  $h$ , a successful perturbation scheme on the original picture. This is obtained by adding the minimum power to generate a successful perturbation on the picture of that object on transparent background, and the power to change the original prediction to the transparent-background prediction. Then, the  $c$ -Eval versus object coverage curve on the transparent background instance will provide us a guarantee on the object coverage of explainers.

Let us consider the picture instance  $x_0$  of object  $o$  in a removed background  $b_0 = \emptyset$  (Fig. 3b). We assume the classifier predicts  $x$  and  $x_0$  correctly, i.e.  $\arg \max_{1 \leq j \leq m} y_j = \arg \max_{1 \leq j \leq m} y_{0j} = j_o$ , where  $\{y_{01}, \dots, y_{0n}\} = \mathbf{y}_0 = f(x_0)$ . Noting that even though  $\arg \max_{1 \leq j \leq m} y_{0j} = \arg \max_{1 \leq j \leq m} y_j$ ,  $\mathbf{y}_0$  is different from  $\mathbf{y}$  in general. The minimum power to successfully perturb  $x$  must be upper bounded by the perturbing power on  $x$  to change the prediction  $f(x)$  into  $f(x_0)$  and then successfully perturb  $f(x_0)$ . Specifically, we have:

$$c_{f,x}(e_x) \leq c_{f,x_0}(e_x) + P_0, \quad (3)$$

where  $c_{f,x}(e_x)$  (or  $c_{f,x_0}(e_x)$ ) is defined as the minimum power to successfully perturb instance  $x$  (or  $x_0$ ) without changing the elements of  $x$  (or  $x_0$ ) in  $e_x$ .  $P_0$  is the necessary power of perturbation to change the prediction from  $f(x)$  to  $f(x_0)$ , i.e.  $f(x + \delta) = f(x)$ . The demonstration of the special case when  $\mathbf{y} = \mathbf{y}_0$ , i.e.  $P_0 = 0$ , is shown in Fig. 3.

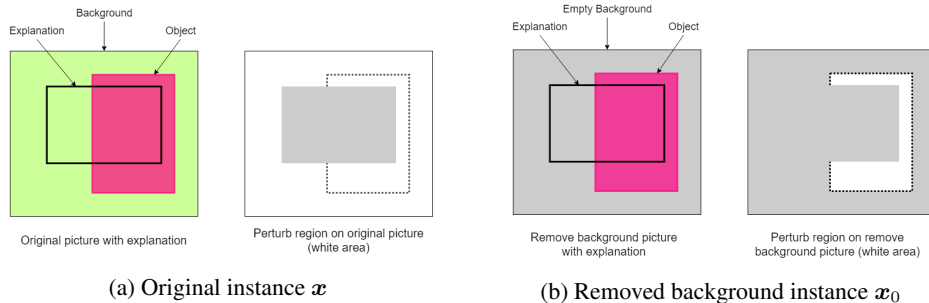


Figure 3: Perturbation on original picture  $x$  and removed background instance  $x_0$ : If the prediction of the original  $x$  and the removed background  $x$  is the same, i.e.  $\mathbf{y} = \mathbf{y}_0$ , the power to perturb the original picture  $c_{f,x}(e_x)$  is upper-bounded by the power to perturb the removed background picture  $c_{f,x_0}(e_x)$  since the perturbing region of the later is a subset of that region in the former.

Let us consider the term  $c_{f,x_0}(e_x)$  on the right hand side of (3). Since  $x_0$  has an empty background,  $x_0$  contains no elements outside  $o$ , we have  $c_{f,x_0}(e_x) = c_{f,x_0}(e_x \cap o)$ . Combining with the facts that  $e_x \cap o \subseteq o$  and  $\text{cov}(e_x \cap o) = \text{cov}(e_x)$ , we obtain the following bound:

$$c_{f,x_0}(e_x) = c_{f,x_0}(e_x \cap o) \leq \max_{s \subseteq o} c_{f,x_0}(s) \quad \text{s.t. } cov(s) \leq cov(e_x). \quad (4)$$

We denote the following coverage to  $c$ -Eval function  $C_{f,x_0}(\Gamma)$  as follows

$$C_{f,x_0}(\Gamma) = \max_{s \subseteq o, |s| \leq \Gamma|o|} c_{f,x_0}(s), \quad \Gamma \in [0, 1]. \quad (5)$$

Intuitively, the solution  $s_0 \subseteq o$  that results in  $C_{f,x_0}(\Gamma)$  is the elements in object  $o$  that, by keeping them unchanged, the required power to successfully perturb  $x_0$  increases the most. Additionally,  $\Gamma$  can be interpreted as the coverage ratio of object  $o$ . In the next theorem, we will show that  $C_{f,x_0}(\Gamma)$  is monotone non-decreasing function of  $\Gamma$ .

**Theorem 1.** *Given a classifier  $f$  and an instance of empty background  $x_0$ ,  $C_{f,x_0}(\Gamma)$  is monotone non-decreasing function of  $\Gamma \in [0, 1]$ .*

*Proof.* We consider  $0 \leq \Gamma_1 \leq \Gamma_2 \leq 1$  and denote  $s_0^{(1)}$  and  $s_0^{(2)}$  the solutions of (5) when  $\Gamma$  equals to  $\Gamma_1$  and  $\Gamma_2$  respectively. For all subset  $s$  of  $o$  such that  $|s| \leq \Gamma_2 - \Gamma_1$ ,  $s \cap s_0^{(1)} = \emptyset$ , we have

$$C_{f,x_0}(\Gamma_1) = c_{f,x_0}(s_0^{(1)}) \leq c_{f,x_0}(s_0^{(1)} \cup s) \leq c_{f,x_0}(s_0^{(2)}) = C_{f,x_0}(\Gamma_2), \quad (6)$$

where the first inequality in (6) is because the minimum power to successfully perturb outside  $s_0^{(1)}$  is greater than the minimum power to successfully perturb outside  $s_0^{(1)} \cup s$ . The second equality is from  $s_0^{(1)} \cup s \subseteq o$  and  $cov(s_0^{(1)} \cup s) = cov(s_0^{(1)}) + cov(s) \leq \Gamma_2$ .  $\square$

Theorem 1 allows us to define the inverse function of  $C_{f,x_0}(\Gamma)$ :

$$C_{f,x_0}^{-1}(\gamma) = \arg \min_{\Gamma} C_{f,x_0}(\Gamma), \quad \text{s.t. } C_{f,x_0}(\Gamma) = \gamma. \quad (7)$$

The inverse function  $C_{f,x_0}^{-1}(\gamma)$  computes the guarantee object coverage ratio of object  $o$  for any explanation on  $x_0$  that result in  $c$ -Eval of  $\gamma$ . From the definition of  $C_{f,x_0}^{-1}(\gamma)$  and the monotone property of  $C_{f,x_0}(\Gamma)$ , it is clear that  $C_{f,x_0}^{-1}(\gamma)$  is monotone increasing. To analyze the coverage of  $e_x$ , we apply the inverse  $c$ -Eval to coverage ratio function  $C_{f,x_0}^{-1}(\gamma)$  as into (3):

$$\begin{aligned} c_{f,x}(e_x) \leq c_{f,x_0}(e_x) + P_0 &\implies c_{f,x_0}(e_x) \geq c_{f,x}(e_x) - P_0 \\ \implies C_{f,x_0}(cov(e_x)) \geq c_{f,x}(e_x) - P_0 &\implies cov(e_x) \geq C_{f,x_0}^{-1}(c_{f,x}(e_x) - P_0), \end{aligned} \quad (8) \quad (9)$$

where (9) is from the definition of (5) and by applying the inverse function  $C_{f,x_0}^{-1}(\gamma)$  to both sides. This final result gives us the object guarantee Theorem based on  $c$ -Eval:

**Theorem 2.** *For any explanation  $e_x$ , it must cover at least  $C_{f,x_0}^{-1}(c_{f,x}(e_x) - P_0)$  portion of the predicted object in the image  $x$ .*

To demonstrate Theorem 2, we construct the  $C_{f,x_0}(\Gamma)$  function for an image instance in Fig. 4. First, we extract the object from the original picture to obtain the removed background instance. For each coverage ratio, we search for the optimal mask on the subject such that, by keeping all pixels in the mask unchanged, the minimum power to obtain a successful perturbation is maximized. The exact computation of this mask requires the perturbation on all features, which is infeasible for most image instances. To reduce the complexity, we segment the picture into super-pixels and return the solution as subset of super-pixels. The searching algorithm for this subset is conducted by randomization and greedy selection. For each object coverage ratio,  $c$ -Eval of the corresponding mask is computed and plotted in log-scale as shown in Fig. 4. We also generate the LIME and SHAP explanations on this sample picture with different number

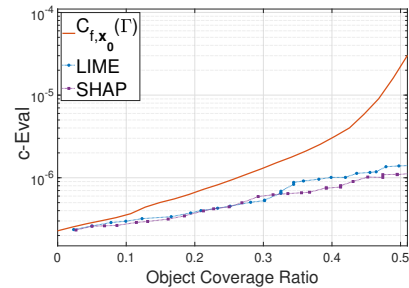


Figure 4:  $c$ -Eval to object coverage of explainers and  $C_{f,x_0}(\Gamma)$  curve. For  $P_0$  small, the explainers stay below  $C_{f,x_0}(\Gamma)$  and  $c$ -Eval establish the object coverage guarantee of explainers.

of explaining features. After obtaining the perturbation on the explanations, we compute their corresponding object coverage ratios and plot the results. Since SHAP explanation is in the form of pixels' weight instead of segments, we generate the segment representation of SHAP as the segments with maximum sum weights. The detailed algorithm for object coverage curve is discussed in Appendix A.

From Fig. 4, Theorem 2 let us claim that any explanation  $e_x$  with  $c$ -Eval greater than  $10^{-6}$  should cover at least 25% of the predicted object in the original picture  $x$ . In fact, with  $c$ -Eval of  $10^{-6}$ , LIME and SHAP explanations in this case cover about 35% and 45% of the predicted object.

#### 4 c-Eval plot: a low-complexity indicator for explanations quality

As can be seen in the previous section,  $c$ -Eval implies that the corresponding explanation must cover a certain portion of the predicted object in the examined instance; however, this theoretical coverage bound might be computationally challenging in some cases. In analyzing the prediction of average size images, the number of features might be too large for us to find the optimal mask (5). To overcome this challenge, we provide in this section another heuristic approach, called  $c$ -Eval plot, to evaluate the quality of explainers. Given an explainer and a prediction, we vary the number of features  $k$  returned by the explainer, compute the explanation and determine the corresponding  $c$ -Eval. In the followings, we demonstrate that plotting the obtained  $c$ -Eval and examining the curve will help us assess the explainers' quality.

Our first observation is that most modern feature-based explainers will assign an importance weight to each feature before outputting the final explanations [2–5]. In other words, when an explainer provides explanation  $e_x$  containing  $k$  features, those  $k$  features are the features with highest importance weights. For LIME [3], user can choose  $k$  to adjust the compactness of the explanation. SHAP [2] and GCam [4], on the other hand, provide the complete importance map on all pixels of the examined picture. To unify all these representations of different explainers, we denote the collection of explanations by an explainer on instance  $x$  as  $\{e_x^k\}_{k=0}^n$ , where  $k$  indicates how many features are included in the explanation. For simplicity, we assume that all features have distinct importance weights. Thus, we obtain  $e_x^k \subseteq e_x^{k+1}$  for all  $0 \leq k \leq n$ .

Assuming we have an ideal scoring machine  $\mathcal{S}$  to evaluate the explanation quality for each budget  $k$ . For example, when  $k = 1$ , each explainer is expected to provide an answer as a picture segment. Then, the scoring machine will give score  $\mathcal{S}(e_x^1)$  to explanation  $e_x^1$ . The segment will have high score if the machine thinks it correctly explains why the classifier made the corresponding prediction. The segment will be given a lower score if it is less relevant to the prediction. For  $k = 0$  and  $k = n$ , the scores for all explainers are equal since their outputs are the same. Thus, a good explainer is the one with high scores  $\mathcal{S}(e_x^k)$  for  $k$  between 1 and  $n - 1$ . We assume that  $\mathcal{S}(e_x^i) \leq \mathcal{S}(e_x^j)$  if  $i \leq j$  for each explainer since the more segments allowed, the more information the explanation can contain. If we plot the scoring  $\mathcal{S}(e_x^k)$  as a function of  $k$  for two explainers, the starting points and ending points of them intersect. The curve for the better explainer is on top of the other.

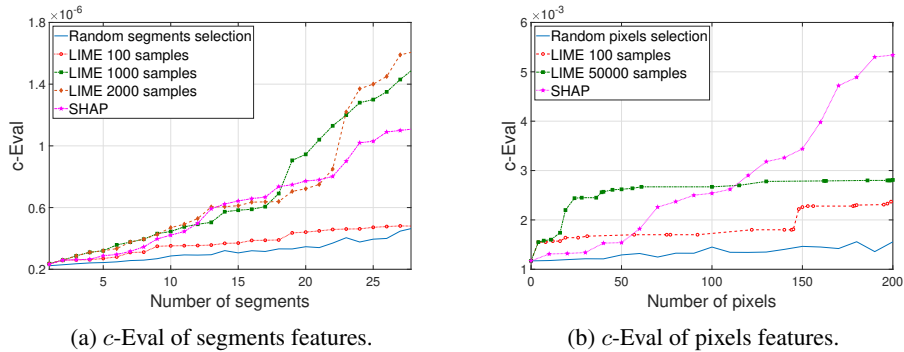


Figure 5:  $c$ -Eval of random selection, LIME with different sample rates and SHAP. The random selection and LIME-100 are expected to be worse than others and  $c$ -Eval plot reflect that expectation.

In the followings, we heuristically show that the  $c$ -Eval behaves similarly to the score given by the ideal machine  $\mathcal{S}$ . Noting that any successful perturbation on  $x \setminus e_x$  should be a successful perturbation without changing any components in any subsets of  $e_x$ . As a result, given a collection  $\{e_x^k\}_{k=0}^n$ , from the definition of  $c$ -Eval, we have  $c_{f,x}(e_x^k) \leq c_{f,x}(e_x^{k+1})$  for all  $k$ . In addition, for all explainers,

$c_{f,x}(e_x^0)$  has the same value since it is the minimum power to successfully perturb  $x$  without any restriction as  $e_x^0 = \emptyset$ . Furthermore, it is straightforward to set  $c_{f,x}(e_x^n) = \infty$  for all explanations because if all features are kept unchanged, there is no way to deviate the classification on that picture. To sum-up, if we plot the sequence  $\{c_{f,x}(e_x^k)\}_{k=1}^n$ , it is a positive non-decreasing sequence starting at  $c_{f,x}(\emptyset)$  and approaching infinity as  $k \rightarrow n$ .

Fig. 5a plots the sequences  $\{c_{f,x}(e_x^k)\}_{k=1}^n$  for random explanation, SHAP explanation and LIME explanation with different number of samplings. We segment the studied picture instance into  $n = 80$  segments. For random explanation, the segments are selected randomly. As we do not have  $e_x^k \subseteq e_x^{k+1}$  for random selection, the curve is not monotonic. For SHAP, we select the segments having the largest sum pixels' weight. In LIME, the sampling size determines how many perturbations are conducted in finding the explanation. The higher the number, the better the explanation [3]. Here, we set LIME samplings size to 100, 1000 and 2000 samples respectively. We denote the corresponding curves as LIME-100, LIME-1000 and LIME-2000. The ideal scoring  $S$  for LIME-100 and random selection are expected to be smaller than the other three. The figure shows a distinct gap in  $c$ -Eval between "good" and "bad" explainers. This result implies we can use  $c$ -Eval plot to evaluate the explainers' quality. We also conduct the same experiments for pixel-wise features in Fig. 5b for an image in MNIST dataset [14]. Obtained result also supports the applicability of  $c$ -Eval in evaluating explainers.

The computation of  $c$ -Eval plot is much more efficient than the object coverage guarantee  $C_{f,x_0}(\Gamma)$ . To obtain the exact  $C_{f,x_0}(\Gamma)$  curve, we not only require the object-only image but also need to determine  $c$ -Eval for all  $2^n$  permutations. On the other hand, we only need to compute at most  $n$  values of  $c$ -Eval to obtain the curve.

## 5 Simulation results

In this section, we provide experiments results of  $c$ -Eval on small gray-scale hand-writing images in MNIST [14] and large color object images in Caltech101 [20]. 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 the power to perturb the image on empty mask  $c_{f,x}(\emptyset)$ . This ratio is also indicated by the notation  $C(e_x)/C_0$  in the legend of each figure. Additionally, since our aim is to examine the usage of  $c$ -Eval in evaluating explainers' quality, we need to have the ground-truths ranking of explainers. These ground-truths are obtained by 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. We provide further discussions and detailed implementations of experimental results in Appendix B, which also includes experiments on CIFAR10 dataset [21].

### 5.1 Simulations on MNIST dataset

In the experiments with MNIST dataset [14], we study 8 different images explainers: LIME [3], SHAP [2], GCam [4], DeepLIFT (DEEP) [5], Integrated Gradients [22], Layerwise Relevance Propagation (LRP) [6], Guided-Backpropagation (GB) [7] and Simonyan-Gradient (Grad) [8].

LIME approximates the importance of each picture segment with a heuristic linear function. The approach in LIME is a specific case of SHAP, which relies on the theoretical analysis of Shapley value in game theory. SHAP explainer 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. Authors of [2] also provide 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. The final studied explainer GCam is an image explainer exploiting the last convolution layer to explain the prediction. Since GCam is not designed for low-resolution images, we expect the explanation quality and the corresponding  $c$ -Eval in MNIST dataset cannot be relatively high.

**Experiments on different models:** Figs. 6a and 6b are the distributions of  $c$ -Evals of 1000 images in MNIST dataset on classifier 1 provided by [2] and classifier 2 provided by [5]. The green lines are the



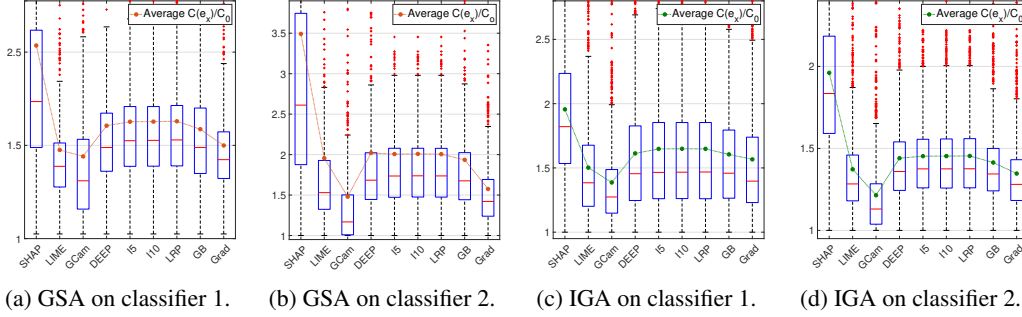


Figure 6: We conduct the experiments for 8 explainers on 1000 images of MNIST dataset. Figure shows the distributions and the averages of  $c$ -Eval for 8 explainers on classifier 1 provided by [2] and on classifier 2 provided by [5].

mean values of the  $c$ -Eval for each explainer. We can see that the behavior of  $c$ -Eval is consistent with the explainer’s performance as we expected. The notation I5 and I10 indicate the Integrated-Gradient method with 5 and 10 interpolations [22]. The result is also consistent with previous attempts of evaluating explainers in [2] and [5]. For the consistency in the behavior of  $c$ -Eval and log-odds function, please see the discussion in Appendix C.

**Experiments on different perturbation schemes:** Figs. 6c and 6d demonstrate the usage of Iterative-Gradient-Attack instead of Gradient-Sign-Attack [19] as in experiments of Figs. 6a and 6b. Comparing the distributions in Fig. 6c to Fig. 6a and Fig. 6d to Fig. 6b, we observe that the relative  $c$ -Eval of explainers are similar on both perturbation schemes. Thus, the computed  $c$ -Evals still reflect the explainers’ performance. The problem of finding optimal perturbation schemes resulting in a good measurement of  $c$ -Eval is not considered in this work; however, the experiments suggest that we can use non-optimal perturbation scheme to obtain reasonable measurement of  $c$ -Eval.

## 5.2 Simulations on Caltech101 dataset

For experiments on large images, we study the performance of LIME [3], SHAP [2], GCam [4], DeepLIFT [5] on 700 images in Caltech101 dataset [20] with the VGG19 classifier [23]. As the first three explainers are designed for medium-size to large-size images, we expect they should outperform DeepLIFT. Furthermore, as discussed above, the results from [2] implies SHAP should perform better than LIME. For the improvement of GCam and the degradation of DeepLIFT from the MNIST dataset and CIFAR10 to the Caltech101 dataset, we suggest that readers focus on the change in quality of explainers from Figs. 9 and 10 to Fig. 11 in Appendix B. The experimental results of  $c$ -Eval in Fig. 7a using Gradient-Sign-Attack and Fig. 7b using Iterative-Gradient-Attack agree with our expectations on explainers’ performance. We also include some examples of explanations and their corresponding  $c$ -Eval in Appendix B to validate the correlation of  $c$ -Eval with the explainers’ quality.

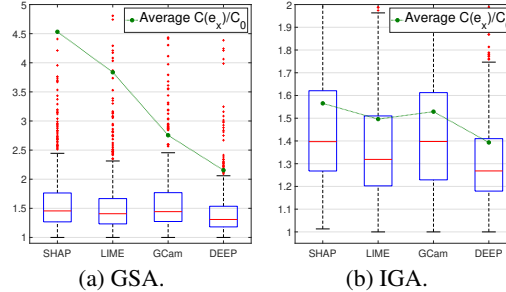


Figure 7: Distribution for  $c$ -Eval of 700 images in Caltech101 dataset for 4 explainers.

## 6 Conclusions

Throughout this research, we introduce the  $c$ -Eval, establish two methods to evaluate feature-based explainers using  $c$ -Eval and conduct extensive experiments on the proposed metric. Nevertheless, there still exists many open questions on the parameters. For example, the discrepancy in the mean values and the distributions of Figs. 7a and 7b suggests that  $c$ -Eval might be sensitive to the perturbation schemes. One question of interest is which perturbation scheme would give us the best measurement of  $c$ -Eval. Additionally, the distributions of  $c$ -Eval in Fig. 6 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.  $c$ -Eval offers us a clear quantification that might shed light into many unanswered questions behind machine learning explainers.



## 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, pp. 115 EP –, Jan 2017. [Online]. Available: <https://doi.org/10.1038/nature21056>
- [2] S. Lundberg and S. Lee, “A unified approach to interpreting model predictions,” *CoRR*, vol. abs/1705.07874, 2017. [Online]. Available: <http://arxiv.org/abs/1705.07874>
- [3] M. T. Ribeiro, S. Singh, and C. Guestrin, ““why should I trust you?”: Explaining the predictions of any classifier,” *CoRR*, vol. abs/1602.04938, 2016. [Online]. Available: <http://arxiv.org/abs/1602.04938>
- [4] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, “Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization,” *CoRR*, vol. abs/1610.02391, 2016. [Online]. Available: <http://arxiv.org/abs/1610.02391>
- [5] A. Shrikumar, P. Greenside, and A. Kundaje, “Learning important features through propagating activation differences,” *CoRR*, vol. abs/1704.02685, 2017. [Online]. Available: <http://arxiv.org/abs/1704.02685>
- [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. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller, “Striving for simplicity: The all convolutional net,” *CoRR*, vol. abs/1412.6806, 2014.
- [8] K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep inside convolutional networks: Visualising image classification models and saliency maps,” *CoRR*, vol. abs/1312.6034, 2013.
- [9] D. Smilkov, N. Thorat, B. Kim, F. B. Viégas, and M. Wattenberg, “Smoothgrad: removing noise by adding noise,” *CoRR*, 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,” *CoRR*, vol. abs/1703.01365, 2017. [Online]. Available: <http://arxiv.org/abs/1703.01365>
- [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] D. Martens and F. Provost, “Explaining data-driven document classifications,” *MIS Q.*, vol. 38, no. 1, pp. 73–100, Mar. 2014. [Online]. Available: <https://doi.org/10.25300/MISQ/2014/38.1.04>
- [14] Y. LeCun and C. Cortes, “MNIST handwritten digit database,” 2010. [Online]. Available: <http://yann.lecun.com/exdb/mnist/>
- [15] Z. C. Lipton, “The mythos of model interpretability,” *CoRR*, vol. abs/1606.03490, 2016. [Online]. Available: <http://arxiv.org/abs/1606.03490>
- [16] B. Kim, E. Glassman, B. Johnson, and J. Shah, “iBCM : Interactive Bayesian case model empowering humans via intuitive interaction,” 2015.
- [17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception architecture for computer vision,” *CoRR*, vol. abs/1512.00567, 2015. [Online]. Available: <http://arxiv.org/abs/1512.00567>
- [18] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” *CoRR*, vol. abs/1412.6572, 2015.
- [19] 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>

- [20] 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.
- [21] A. Krizhevsky, “Learning multiple layers of features from tiny images,” 2009.
- [22] M. Sundararajan, A. Taly, and Q. Yan, “Gradients of counterfactuals,” *CoRR*, vol. abs/1611.02639, 2016. [Online]. Available: <http://arxiv.org/abs/1611.02639>
- [23] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: <http://arxiv.org/abs/1409.1556>
- [24] 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.

## A Experiment on object coverage guarantee bound

In this Appendix, we provide the detailed algorithms for the object coverage guarantee in Section 3.

---

### Algorithm 1: Exact computations of $C_{f,x_0}(\Gamma)$

---

**Input:** Classifier  $f$ , object only image  $x_0$  and perturbation scheme  $h$ .

**Parameter:** Number of segments  $n$ .

**Output:** Sequence of object coverage ratio  $\Gamma$  and the corresponding  $C_{f,x_0}(\Gamma)$ .

---

```

1 Segmentize  $x_0$  into  $n$  segments
2 Filter out the background of  $x_0$  and obtain the object  $o \subseteq x_0$ 
3 Set  $S_0$  be the set of segment indices containing the object  $o$ 
4  $k \leftarrow 0$ 
5 while  $c_{f,x_0}(s) < \infty$  do
6    $s_k = \arg \max_{s \subseteq S_0, |s|=k} c_{f,x_0}(s)$ 
7    $\gamma_k = |s|/|o|$ 
8    $C_{f,x_0}(\gamma_k) = c_{f,x_0}(s_k)$ 
9    $k \leftarrow k + 1$ 
10 end
11 return  $\{\gamma_k\}_k$  and  $\{C_{f,x_0}(\gamma_k)\}_k$ 

```

---

Algorithm 1 shows the procedure to exactly compute  $C_{f,x_0}(\Gamma)$ . First, we segmentize the examined image into  $n$  segment. Then, we collect all segments containing the predicted object and put them into  $S_0$ . For each number of explained features  $k$ , we find the optimal subset  $s$  of  $S_0$  maximizing the  $c$ -Eval under the constraint  $|s| = k$  (line 6). For each  $s$ , we find the perturbation  $h(x_0)$  that successfully changes the classifier prediction while keeping  $s$  unchanged. This computation requires the trial of all subsets  $s$ . After that, we calculate the object coverage ratio and continue the computations until the perturbation scheme  $h$  cannot find any successful perturbation, i.e.  $c_{f,x_0}(s) = \infty$ .

---

### Algorithm 2: Approximation of $C_{f,x_0}(\Gamma)$

---

**Input:** Classifier  $f$ , object only image  $x_0$  and perturbation scheme  $h$ .

**Parameter:** Number of segments  $n$  and number of trials  $\{t_k\}_{k=1}^n$ .

**Output:** Sequence of object coverage ratio  $\Gamma$  and the approximate of  $C_{f,x_0}(\Gamma)$ .

---

```

1 Segmentize  $x_0$  into  $n$  segments
2 Filter out the background of  $x_0$  and obtain the object  $o \subseteq x_0$ 
3 Set  $S_0$  be the set of segment indices containing the object  $o$ 
4  $k \leftarrow 0$ 
5 while  $c_{f,x_0}(s) < \infty$  do
6   Randomly select the collection  $\mathcal{S}$  of  $t_k$  subsets  $s$  such that  $s \subseteq S_0, |s| = k$ 
7    $s_k^1 = \arg \max_{s \in \mathcal{S}} c_{f,x_0}(s)$ 
8    $i_k = \arg \max_{i \in S_0 \setminus s_{k-1}} c_{f,x_0}(s_{k-1} \cup \{i\})$ 
9    $s_k^2 = s_{k-1} \cup \{i_k\}$ 
10   $s_k = \arg \max_{s \in \{s_k^1, s_k^2\}} c_{f,x_0}(s)$ 
11   $\gamma_k = |s|/|o|$ 
12   $C_{f,x_0}(\gamma_k) = c_{f,x_0}(s_k)$ 
13   $k \leftarrow k + 1$ 
14 end
15 return  $\{\gamma_k\}_k$  and  $\{C_{f,x_0}(\gamma_k)\}_k$ 

```

---

For the experiments in Fig. 4 where the number of image segments is relatively large, it is expensive to compute the  $c$ -eval for all subsets  $s$ . As a result, we make a slight modification to the optimization program at line 6 of Algorithm 1 and obtain Algorithm 2. For each  $k \in \{0, \dots, n\}$ , we first randomly select  $t_k$  subsets  $s$  of image segments, add them to collection  $\mathcal{S}$ , compute the corresponding  $c$ -Eval and obtain  $s_k^1$  which is the best  $s$  among  $\mathcal{S}$  (line 7). We also conduct a greedy selection of image segment  $i_0$  and add it to the previous solution  $s_{k-1}$  to obtain  $s_k^2$  (line 8). The solution for iteration  $k$  is the better among  $s_k^1$  and  $s_k^2$ . Specifically, for the experiment in Fig. 4, we use the InceptionV3 classifier [17] and the Gradient-Sign-Attack perturbation scheme [18] for demonstrations.

## B Experiments on MNIST, Caltech101 and CIFAR10

In this appendix, we provide detailed implementations and discussions on experiments in MNIST, Caltech101 and CIFAR10 datasets.

The experiments in the MNIST dataset is conducted in pixel-wise manner. For each image, each explainer except LIME will return 10% of the image as explanation. For LIME, since the algorithm always returns image segments as explanation, we set the returned pixels to be as close to 10% of the image as possible. Another note is that the implementations of LRP are simplified into Gradient $\times$ Input based on the discussion in [5]. With each explanation, we then compute the corresponding  $c$ -Eval and plot them as shown in Fig. 6. The studied classifiers are taken from [2] and [5]. Some examples explanations of images from MNIST are plotted in Fig. 9.

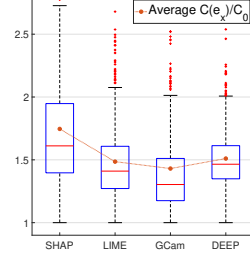


Figure 8: Distributions of  $c$ -Eval on CIFAR10 dataset.

Experiments with the Caltech101 dataset, on the other hand, uses segment-wise features on VGG19 classifier [20]. Since the returned features of many explainers are importance weights of pixels, we need to convert them into a subset of image segments as an explanation for fair comparison. We first segmentize each image and then sum up the importance weights of all pixels inside each segment. We finally select the top  $k$  segments with maximum sum-weight as the segment-wise explanation of the studied explainer. For the results in Fig. 7 each explainer returns the explanation with the number of segments roughly covers about 20% of the original image. The examples of explanation for this case is shown in Fig. 11.

Besides MNIST and Caltech101, we also conduct experiment on the small color image dataset CIFAR10 [21]. The distributions of  $c$ -Eval on 500 images of the dataset and some examples are shown in Figs. 8 and 10. The experimental parameters model and the segmentation procedure are similar to that on Caltech101 dataset. The classifier model in this experiment is the adaptation of VGG on CIFAR10 [24]. Results in Fig. 10 suggest a relative ranking in performance of studied explainers.

Many interesting results and deductions can be drawn out from experiments of three datasets. Besides our previous general analysis presented in Section 5, we would like to point out several key observations as follows:

- Our first comment is about the correlation of  $c$ -Eval and the portion of predicted object captured by explainers in CIFAR10 (Fig. 10) and especially Caltech101 (Fig. 11). Below each image, we report the ratio of  $c$ -Eval  $c_{f,x}(e_x)$  over the power to perturb the original image  $c_{f,x}(\emptyset)$  for normalization. It is clear to us that most explanations containing the essential components of the predicted object have high  $c$ -Eval, which agrees with our theoretical result on the object coverage guarantee in Section 3. For the MNIST dataset, it is non-trivial to assess explainers' quality by a pure observation on Fig. 9. This is also the main reason motivating [5] to propose the log-odds function to evaluate explainers specifically for the MNIST dataset. We have provided the detailed discussions on this matter in Appendix C.
- Our second attention is on the quality of Grad-CAM in three datasets. Since the explainer is designed for fully-connected convolution networks (e.g. VGG) and it exploits the last layer of such network to generate the explanation [4], we expect Grad-CAM performs relatively well in the Caltech101 dataset. For CIFAR10, the adaptation of VGG on CIFAR10 [24] contains only 4 neurons. This specific structure of the model would degrade the explainer performance significantly. Our expectation aligns with the examples shown in Fig. 10 and Fig. 11. The distributions of  $c$ -Eval for two datasets in Fig. 8 and Fig. 7b also reflect those expectations on Grad-CAM.
- 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. 6 are taken directly from the source code of the explainer's paper. On the other hand, 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 discussion 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. A quick glance at SHAP on MNIST in Fig. 9 might suggest that the explainer is worse than some other back-propagation method such as DeepLIFT, Integrated Gradient or Guided-Backpropagation; however, the figure shows that SHAP captures some important features that are overlooked by other methods. Let's consider the explanation of number 4 as an example. SHAP is the only explainer detecting that the area on top of number 4 is important. In fact, this area is actually essential since, if these pixels are white instead of black, the original prediction should be 0 instead of 4. Without the  $c$ -Eval computations, we might solely assess SHAP based on intuitive observations and wrongly evaluate the explainer.

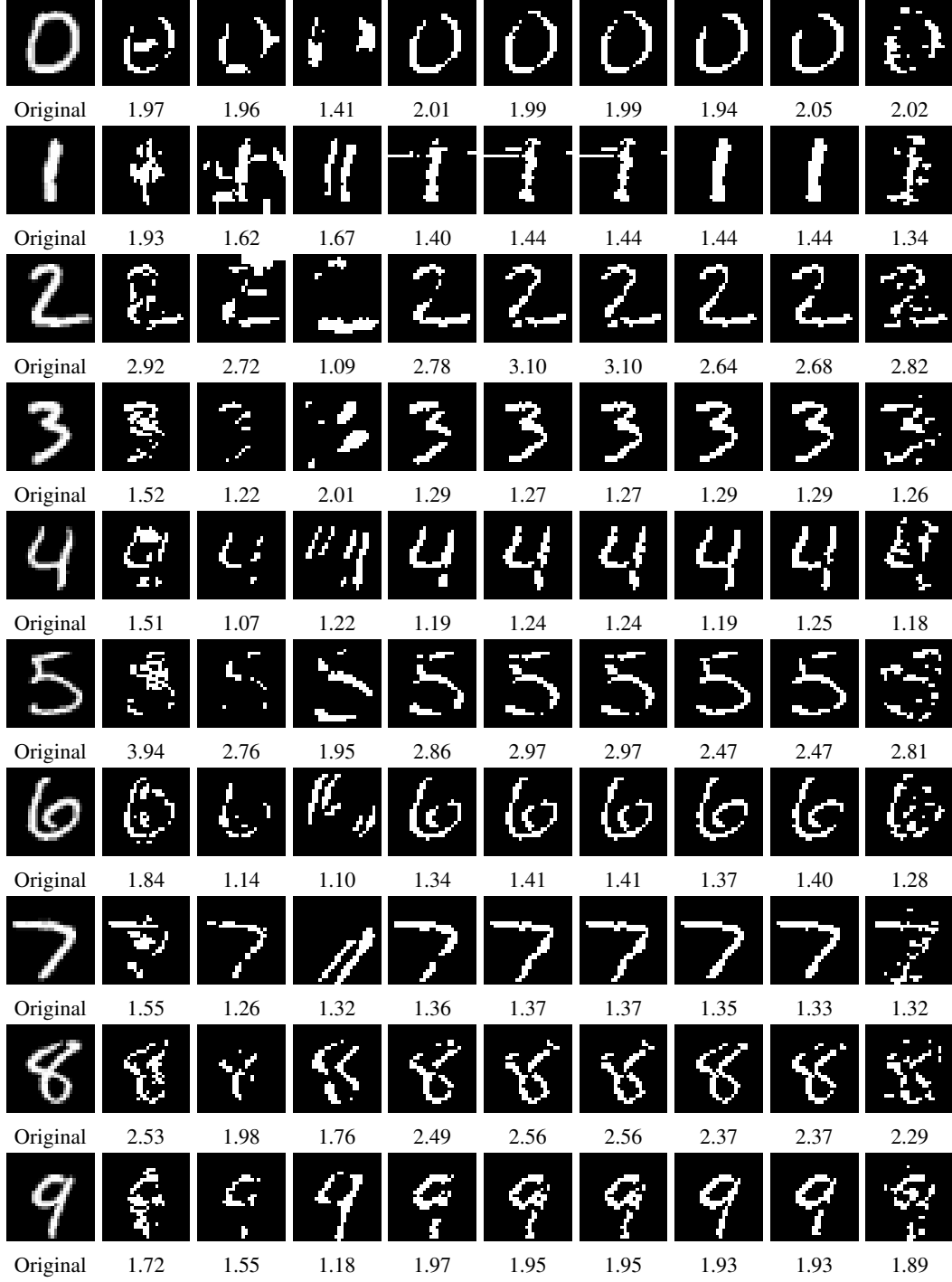


Figure 9: Some examples of explanations and  $c$ -Eval on MNIST. The explainers from left to right: SHAP, LIME, GCam, DeepLIFT, Integrated Gradient with 5 and 10 interpolations, Guided Backpropagation, and Gradient. The number associated with each figure is the ratio  $c_{f,x}(e_x)/c_{f,x}(\emptyset)$ . It is non-intuitive to evaluate these explanations purely by observations. For detailed discussions, please see appendix C.

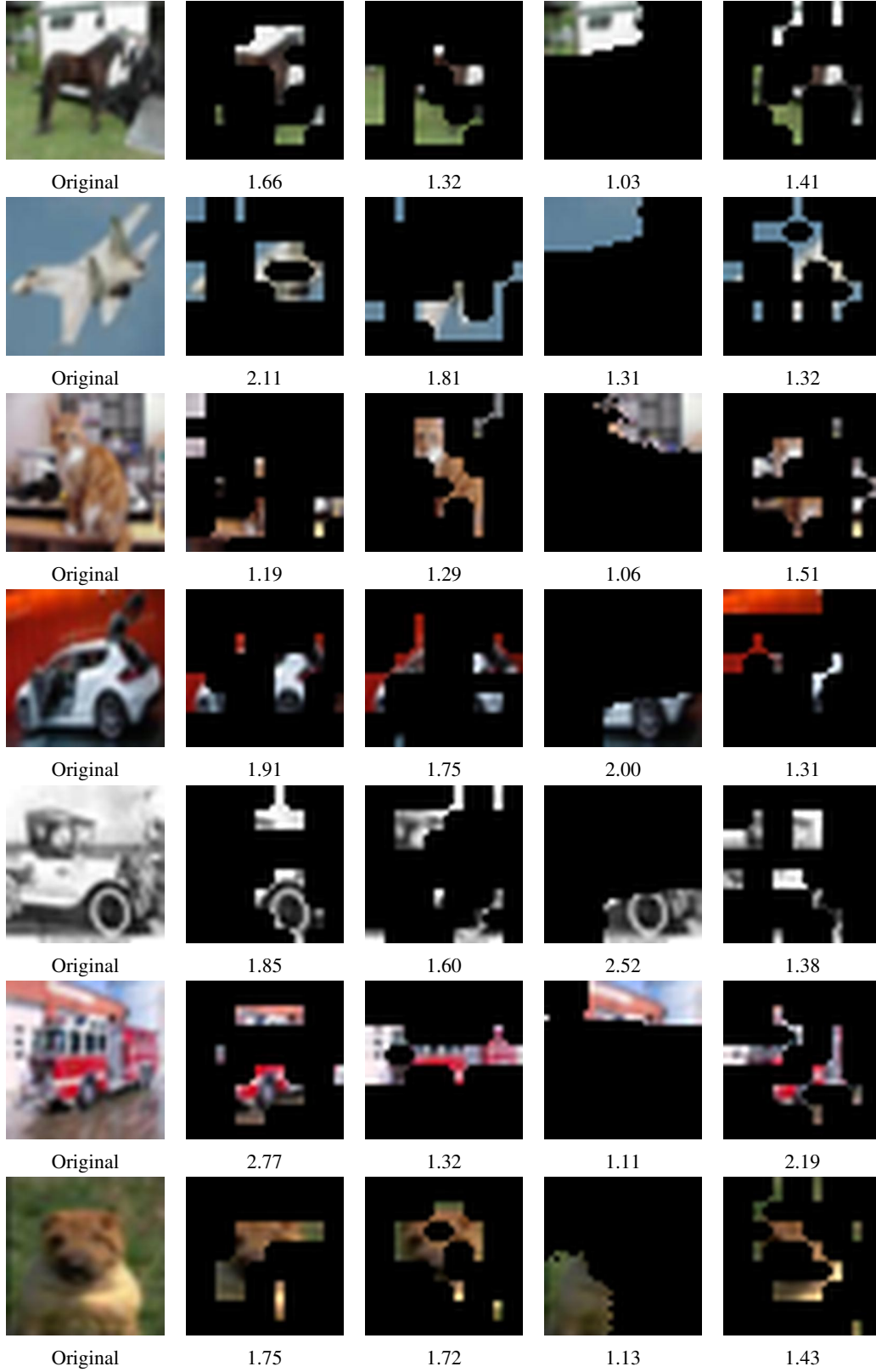


Figure 10: Some examples of Explanations and  $c$ -Eval on CIFAR10. The explainer from left to right: SHAP, LIME, Gcam and DeepLIFT. The number associated with each figure is the ratio  $c_{f,x}(e_x)/c_{f,x}(\emptyset)$ . We observe that most explanations which capture the signature components of the images have relatively high  $c$ -Eval.



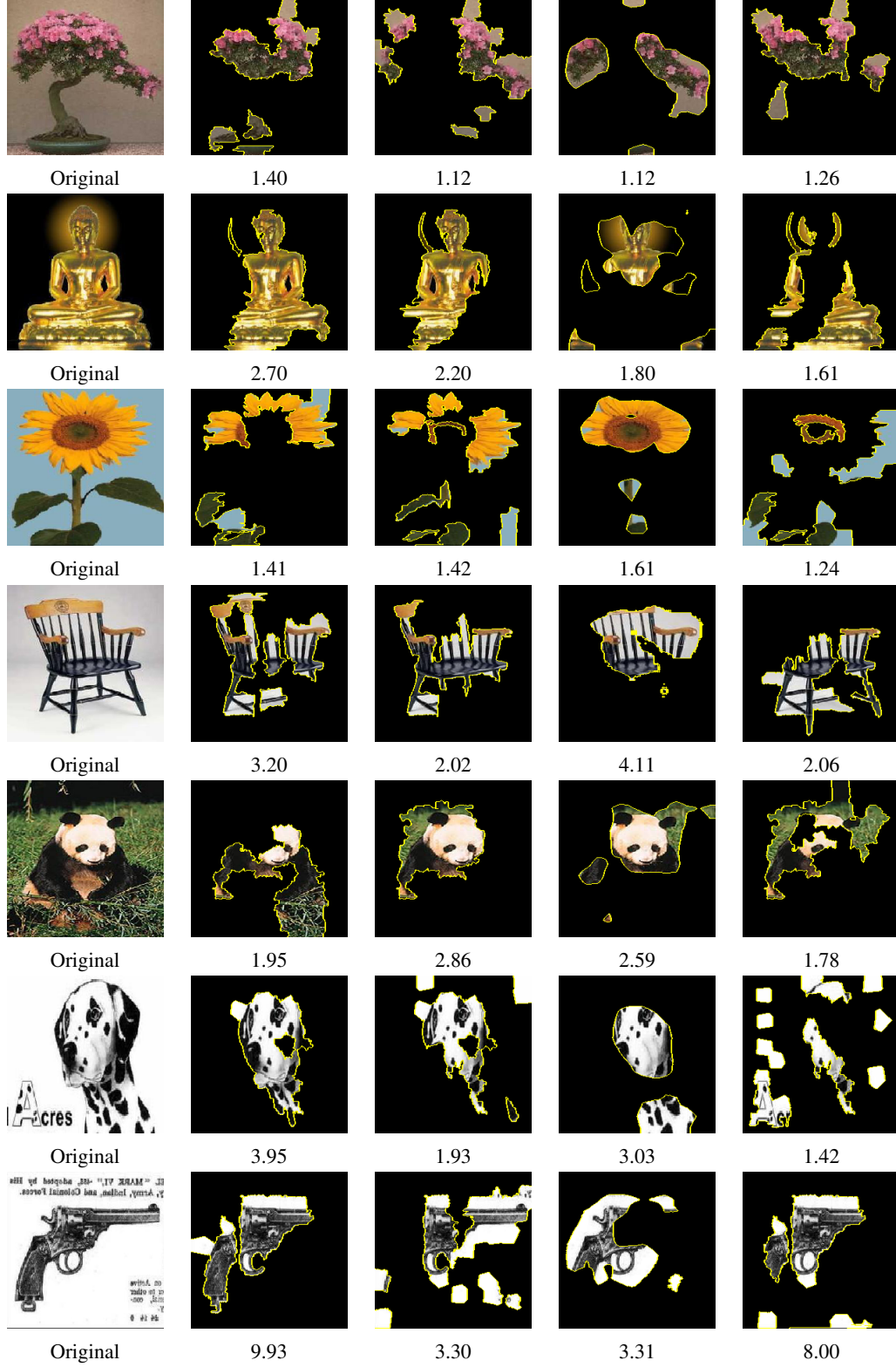


Figure 11: Some examples of Explanations and  $c$ -Eval on Caltech101. The explainers from left to right: SHAP, LIME, GCam and DeepLIFT. The number associated with each figure is the ratio  $c_{f,x}(e_x)/c_{f,x}(\emptyset)$ . We observe that most explanations which capture the signature components of the images have relatively high  $c$ -Eval.

## C Similarity of $c$ -Eval and log-odds functions

To evaluate importance scores obtained by different methods on MNIST dataset, the authors of DeepLIFT [5] 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. All obtained results agree that Guided-Backpropagation and Simonyan-Gradient are inferior to others. Their results also demonstrate that the proposed DeepLIFT is superior in term of log-odds score.

In Fig. 12, we adopt the  $c$ -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 prediction 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.

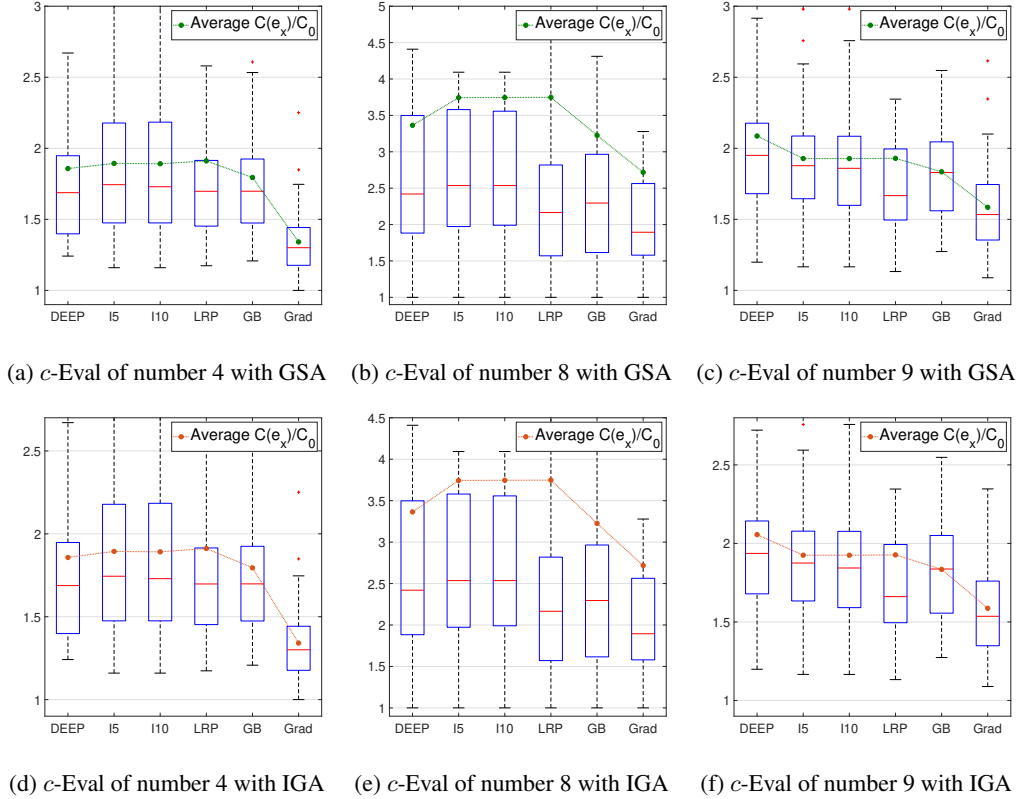


Figure 12: We compute the  $c$ -Eval for 6 explainers on 1000 images of MNIST for number 4,8 and 9 to show the similarity between  $c$ -Eval and log-odds function in [5].