



# Good-looking but Lacking Faithfulness: Understanding Local Explanation Methods through Trend-based Testing

Jinwen He

hejinwen@iie.ac.cn

SKLOIS, IIE, CAS<sup>†</sup>

School of Cyber Security, UCAS<sup>‡</sup>

Beijing, China

Kai Chen\*

chenkai@iie.ac.cn

SKLOIS, IIE, CAS<sup>†</sup>

School of Cyber Security, UCAS<sup>‡</sup>

Beijing, China

Guozhu Meng

mengguozhu@iie.ac.cn

SKLOIS, IIE, CAS<sup>†</sup>

School of Cyber Security, UCAS<sup>‡</sup>

Beijing, China

Jiangshan Zhang

zhangjiangshan@iie.ac.cn

SKLOIS, IIE, CAS<sup>†</sup>

School of Cyber Security, UCAS<sup>‡</sup>

Beijing, China

Congyi Li

licongyi@iie.ac.cn

SKLOIS, IIE, CAS<sup>†</sup>

School of Cyber Security, UCAS<sup>‡</sup>

Beijing, China

## ABSTRACT

While enjoying the great achievements brought by deep learning (DL), people are also worried about the decision made by DL models, since the high degree of non-linearity of DL models makes the decision extremely difficult to understand. Consequently, attacks such as adversarial attacks are easy to carry out, but difficult to detect and explain, which has led to a boom in the research on local explanation methods for explaining model decisions. In this paper, we evaluate the faithfulness of explanation methods and find that traditional tests on faithfulness encounter the random dominance problem, *i.e.*, the random selection performs the best, especially for complex data. To further solve this problem, we propose three trend-based faithfulness tests and empirically demonstrate that the new trend tests can better assess faithfulness than traditional tests on image, natural language and security tasks. We implement the assessment system and evaluate ten popular explanation methods. Benefiting from the trend tests, we successfully assess the explanation methods on complex data for the first time, bringing unprecedented discoveries and inspiring future research. Downstream tasks also greatly benefit from the tests. For example, model debugging equipped with faithful explanation methods performs much better for detecting and correcting accuracy and security problems.

## CCS CONCEPTS

• Security and privacy → Software and application security.

## KEYWORDS

Deep learning; Local explanation; Faithfulness; Security task

\* Corresponding author.

† Institute of Information Engineering, Chinese Academy of Sciences.

‡ University of Chinese Academy of Sciences.



This work is licensed under a Creative Commons Attribution International 4.0 License.

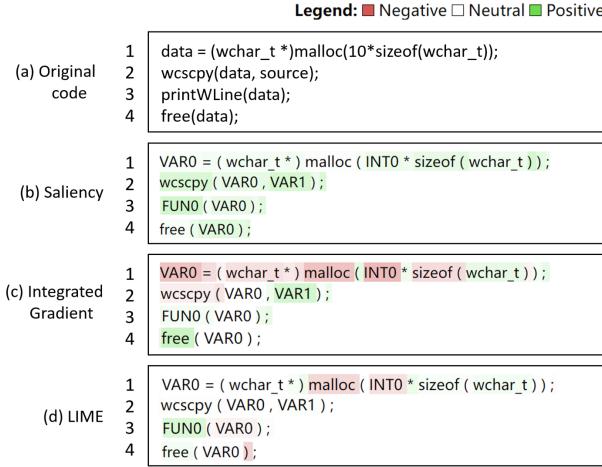
## ACM Reference Format:

Jinwen He, Kai Chen\*, Guozhu Meng, Jiangshan Zhang, and Congyi Li. 2023. Good-looking but Lacking Faithfulness: Understanding Local Explanation Methods through Trend-based Testing. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23), November 26–30, 2023, Copenhagen, Denmark*. ACM, New York, NY, USA, 15 pages. <https://doi.org/10.1145/3576915.3616605>

## 1 INTRODUCTION

In the past ten years, with rapid advances in the field of deep learning (DL), data-driven approaches have drawn lots of attention. They have made great progress in many fields, including computer vision [20, 35], speech recognition [17, 56], natural language processing [50, 57], etc. One of the main benefits of data-driven approaches is that, without needing to know a theory, a machine learning algorithm can be used to analyze a problem using data alone. However, on the other side of the coin, DL models are hard to explain without the theory. Neither can researchers understand why the DL models make a decision. A well-known problem is adversarial examples (AEs), which mislead a DL model by adding human-imperceptible perturbations to the natural data [16]. These perturbations are imperceptible by humans, but impact the decision of the model. To fill the gap between model decisions and human cognition, researchers develop various techniques to explain the prediction results [47, 52, 53]. Obviously, an ideal technique should explain a model's predictions in a *human-understandable* and *model-faithful* manner [29, 63]. That is, the explanation should be meaningful to humans and correspond to the model's behavior in the vicinity of the instance being predicted. The risks of deep learning models further propel the advance of explanation methods, which are popularly used to build secure and trustworthy models [10], such as model debugging [4, 66], understanding attacks [51, 59] and defenses [46] of DL models.

In this paper, we compare popular local explanation methods theoretically and experimentally. Specifically, we implement ten typical methods for comparison. Figure 1 compares the results of Saliency map [53], Integrated Gradient [55] and LIME [47] on a vulnerability detection model trained with VulDeePecker dataset [38]. The contribution of “wcscpy” in the second line differs among the three explanation methods. In Figure 1(b), “wcscpy” has positive

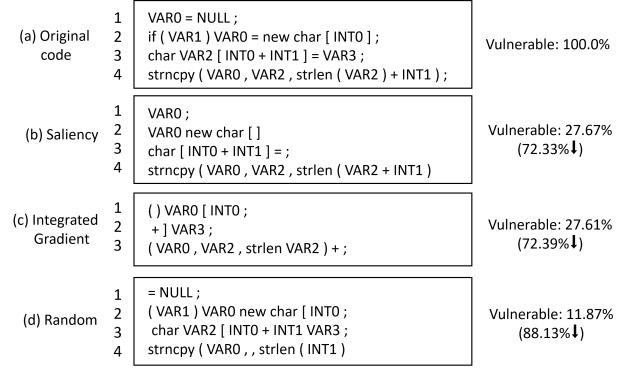


**Figure 1: The importance of words identified by three explanation methods. The darker the color, the higher the contribution score.**

contribution, while in Figure 1(c), “wcscpy” has negative contribution. In Figure 1(d), “wcscpy” has almost no contribution. It is observed that the similarity between the results of different explanation methods is small. Thus, it is highly needed to assess the faithfulness of explanation methods, which is also highly challenging. The main difficulty lies in the lack of ground truth, where contemporary assessments cannot accurately determine the consistency of the explanation with model prediction. Most of these methods rely on the hypothesis to assess explanations that the perturbations imposed to more important features can positively make a larger change to the model prediction. However, this hypothesis suffers from one significant limit, undermining the faithfulness assessment. This limit is dubbed as *random dominance*.

**Random dominance in model explanation.** Take the assessment method— feature reduction [9, 11, 19, 61] as an example, where the difference in prediction scores is measured when important features of the input are deleted. In Figure 2, deleting the outputs by Saliency (Figure 2(b)) lowers the prediction score by 72.33%, and deleting Integrated Gradient’s output (Figure 2(c)) reduces the score by 72.39%. Figure 2 shows the remaining features after removing important features. From the results, the important features tagged by the two methods are very different, but the prediction scores drop a lot for both methods. Surprisingly, if we randomly delete 20% of the input (Figure 2(d)), the score can be reduced by 88.13%, even larger than the two explanation methods. The random method can never be a good explanation.

To solve the problem, we design three new trend tests for explanation assessment: the evolving-model-with-backdoor test (EMBT), partial-trigger test (PTT), and evolving-model test (EMT). Instead of destructing important features, we gradually evolve either a model or a sample, and form a series of test pairs  $\langle model, sample \rangle$ . It enables the models and samples to stay in distribution since the model can continuously learn from the samples during evolution, and the evolution of samples is limited within the cognition scope of the model. We employ the probability and loss function as an indicator to quantify model behaviors and then calculate the correlation with explanation results. Based on these trends, we perform



**Figure 2: The percentage of score decline after removing 20% of the most important or randomly selected words. The random method shows the most significant drop in the prediction score.**

extensive evaluations and analysis of various explanation methods through trend tests and traditional tests. Specifically, we explore the following research questions:

- **RQ1:** How well do the traditional tests work? What are the advantages of trend tests over traditional tests? (See Section 4.2)
- **RQ2:** What factors affect the faithfulness of explanation methods? (See Section 4.3)
- **RQ3:** Do downstream applications such as model debugging work better when using the explanation method chosen by trend tests? (See Section 5)

Through the evaluation, we have the opportunity to assess the explanation methods and gain unprecedented findings. We find that all explanation methods seem to be unable to handle complex data, as indicated by traditional assessment tests. However, our newly designed tests report that some methods (e.g., Integrated Gradient [55] and Integrated SmoothGrad-Squared [54, 55]) can work well. The reason is mainly due to the random dominance problem existing in the traditional tests, which leads to the wrong results of the evaluation report. Furthermore, model complexity seems less important to the explanation methods’ faithfulness than data complexity; but the parameters used by the explanation methods are essential. Some researchers are in favor of the parameters that can generate more explainable features (to humans) but ignore faithfulness. Our trend tests can address this problem by suggesting the most suitable parameters from candidate ones, resulting in the best faithfulness. Moreover, trend tests are applicable to multiple types of models for various tasks, such as images, natural language, security applications, etc. Finally, we demonstrate the effectiveness of trend tests using a popular downstream application, model debugging. For a given DL model, trend tests recommend explanation methods with higher faithfulness to better debug the model, making it secure and trustworthy.

**Contributions.** Our main contributions are as follows:

- We develop three novel trend tests (EMBT, PTT, and EMT) to handle the random dominance problem. They are experimentally proven to be effective in measuring the faithfulness of an explanation method and getting rid of the random dominance problem. All the code and the full version are released for further research: <https://github.com/JenniferHo97/XAI-TREND-TEST>.

- Through the experiments, we identify the limitations of previous assessment methods and quantify the influence of multiple factors (*i.e.*, data complexity, model complexity, parameters) over explanation results.
- We demonstrate that trend tests can recommend more faithful explanation methods for model debugging and thus better detect spurious correlations in DL models.

## 2 BACKGROUND

### 2.1 Explanation on DNN

The high degree of non-linearity of DL models makes it difficult to understand the decision, so security cannot be guaranteed [16]. Such dilemma motivates research on explanation techniques for DL models [37, 70], aiming to explain DL models' decisions [4] and understand adversarial attacks [12, 59] as well as defenses [69], thereby paving the way for building secure and trustworthy models. Explanation methods can be categorized as global explanation and local explanation in terms of the analysis object [10]. In this paper, we focus on local explanation methods. Without loss of generality, we define the explanation method for input as follows.

**DEFINITION 1. (*Local Explanation*)** Given a model  $\mathcal{F} : \mathcal{X} \rightarrow \mathcal{Y}$ , an explanation method  $I : (\mathcal{X}, \mathcal{F}) \rightarrow \phi$ . For any test input  $\mathbf{X}$ , the explanation method gives the importance score  $\phi$  for each feature of  $\mathbf{X}$ , where  $\phi$  has the same dimensions as  $\mathbf{X}$ .

Assuming that  $\{x_1, x_2, \dots, x_n\}$  is the feature set of instance  $\mathbf{X}$  and  $\{\phi_1, \phi_2, \dots, \phi_n\}$  is the importance score set of the explanation  $\phi$ ,  $x_i$  is important for the explanation if  $\phi_i \geq \epsilon$  where  $1 \leq i \leq n$  and  $\epsilon$  is often empirically configured. Local explanation methods can be either white-box or black-box methods. If one explanation method is dependent on the hyper-parameters and weights of the model, it is a white-box method. Otherwise, it is a black-box method. Saliency map [53] is a typical white-box method, which computes gradients of the input. Although simple and easy to implement, the Saliency map suffers from the gradient saturation problem and is sensitive to noise. Integrated gradient (IG) [55] moderates the gradient saturation problem by considering the straight-line path from the baseline to the input and computes the gradients at all points along the path. SmoothGrad [54] tries to reduce the sensitivity of the gradient by adding Gaussian noise to the input and then calculating the average of the gradients. SmoothGrad-Squared (SG-SQ) [23], VarGrad (VG) [2] and Integrated SmoothGrad-Squared (SG-SQ-IG) [23] are common variants of the above methods. Deep Learning Important FeaTures (DeepLIFT) [52] alleviates the gradient saturation by using the difference between the input and the reference point to explain the importance of input features. The black-box methods are perturbation-based. Kernel SHAP [41] and LIME [47] mutate the input randomly. LIME [47] leverages superpixel segmentation [1] to improve efficiency in image tasks. Occlusion [64] uses a moving square to generate perturbed input. Occlusion [64] directly uses the target classification probability as the metric. The lower the probability caused by the mutated input, the more important the features. Based on the local linearity assumption of the neural network decision boundary, LIME [47] trains a surrogate linear model using the perturbed data and labels.

The weights of the linear model reflect the importance of the feature. SHAP [41], derived from cooperative game theory, calculates Shapley values as feature importance.

### 2.2 Relationship between explanations, models and humans

An explanation system usually includes the interaction between explanation methods, models, and humans. Prior work that assesses faithfulness falls into two types: human-understandable and model-faithful. The human-understandable assessments focus on the correlation between explanation methods and human cognition [29]. Unfortunately, explanation methods cannot reveal all the knowledge learned by the model precisely. Therefore, it has not yet reached the stage where we can assess the correlation between explanation methods and human cognition. Under such circumstances, we should evaluate the explanation methods in a model-faithful way. The model-faithful assessments focus on the correlation between the explanation method and the model [19]. A common way is to mask some important features tagged by the explanation method and then observe the decline in the model prediction probability. The more the probability decreases, the more important the masked features are. However, randomly masking some features may also cause a significant decrease in model prediction probability. We refer to this as the random dominance problem. To overcome this problem, we propose trend tests, which use in-distribution data and are applicable in more scenarios. After the model-faithful assessment, the user can select a faithful explanation method to explain the model, fix the bias and improve the security and trustworthiness of the model. Ultimately, consistency in explanation methods, model decisions, and human cognition can be achieved.

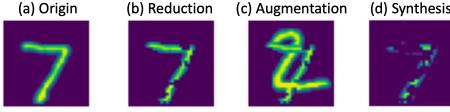
## 3 DESIGN OF FAITHFULNESS TESTS

In this section, we first provide a high-level definition of faithfulness. Then we briefly introduce traditional evaluation methods and design three trend-based tests, *i.e.*, the evolving-model-with-backdoor test (EMBT), partial-trigger test (PTT), and evolving-model test (EMT), to assess the faithfulness of explanation methods.

### 3.1 Problem Definition

A local explanation is faithful if its identified features in the input are what the model relies on for making the decision. However, it is non-trivial to evaluate the faithfulness of an explanation method, as indicated in Figure 1 and 2. The formal definition of faithfulness varies across studies [15, 19]. In this section, we first review the definition of the traditional faithfulness tests from previous work and then present our new trend tests in the next section. Below we use  $\rho$  to denote faithfulness.

**Traditional Faithfulness Tests.** There are three common tests for the local explanation, *i.e.*, synthesis test, augmentation test and reduction test [19, 61]. These tests are widely used as SOTA methods in recent research [14, 15]. The intuition of these tests is to modify an input guided by explanation results and observe the change of the target label's posterior probability by the model, *i.e.*,  $\mathcal{F}_{\text{target}}(\mathbf{X})$ . In the synthesis test, we only retain the important features  $\hat{\mathbf{X}}$  (*i.e.*,



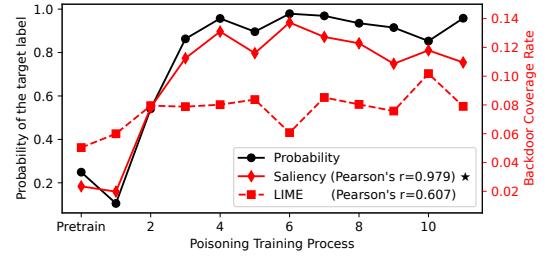
**Figure 3: Examples of synthesis test, augmentation test and reduction test. Features with the top 10% importance scores tagged by the explanation method are important features.**

$\{x_i \mid \phi_i \geq \epsilon\}$ ) of the test sample  $X$  marked by the explanation methods and add them into an all-black image  $X'$  to form a synthetic sample. Then the difference of target label scores between the synthesis test sample and the all-black image could indicate the faithfulness of the explanation methods, denoted by  $\rho_{syn}$ . This can be computed as:  $\rho_{syn}(\mathcal{F}, I) = \mathcal{F}_{target}(X' \oplus \hat{X}) - \mathcal{F}_{target}(X')$ , where  $\oplus$  denotes element-wise addition. Figure 3(a) and (b) show an example of the original test sample and the corresponding synthesis test sample, respectively. Intuitively,  $\rho_{syn}$  will increase after important features are added to the all-black image. In the augmentation test, we randomly select an augmentation sample  $X''$  with a different label from the test samples from the test set. Then we add  $\hat{X}$  to the augmentation sample (see Figure 3(c)) and observe the change of the prediction score:  $\rho_{aug}(\mathcal{F}, I) = \mathcal{F}_{target}(X'' \oplus \hat{X}) - \mathcal{F}_{target}(X'')$ . If important features are accurately recognized,  $\rho_{aug}$  is expected to increase. In the reduction test, we remove important features from the test sample (see Figure 3(d)) and observe:  $\rho_{red}(\mathcal{F}, I) = \mathcal{F}_{target}(X \ominus \hat{X}) - \mathcal{F}_{target}(X)$ , where  $\ominus$  denotes element-wise subtraction. In the reduction test,  $\rho_{red}$  is expected to decrease if the explanation method accurately tags the important features.

### 3.2 Trend-based Faithfulness Tests

The main problem of traditional tests is the random dominance phenomena, which makes the random baseline too high and invalidates the tests. To solve this problem, we design three trend tests. The intuition is: instead of using features to mutate samples, we generate a set of samples with a certain “trend” with natural and backdoor data. Then we let the explanation methods mark important features and check whether the features follow the trend. By measuring the correlation, we can assess the faithfulness of explanation methods.

**Evolving-Model-with-Backdoor Test (EMBT)** To explain a given model, EMBT adds a backdoor to the pre-trained model through incremental training and records the intermediate models in the training process [18]. The probability of the backdoor attack’s target label forms a trend. We assume that the model learns at least some of the backdoor features. During backdoor training, the explanation results should show a trend of paying more attention to the location of the backdoor features. EMBT records the intermediate model in every  $c$  epochs during training. Then we get a set of models  $M = \{M_0, \dots, M_n\}$ . The model  $M_0$  is the pre-trained clean model, and  $M_i$  is the intermediate model generated in the epoch  $i \times c$ . For a given input, EMBT stamps the trigger on the input and measures the probability of the target label on  $M$ . Suppose the result is  $P_{target} = \{P_0, P_1, \dots, P_n\}$ . The black line in Figure 4 shows how the probability of the target label changes during the poisoning training. Later, EMBT uses an explanation method to mark the important features on  $M$ . For each model  $M_i$ , we can calculate the



**Figure 4: EMBT example. Saliency’s backdoor coverage is more related to ResNet18’s backdoor accuracy, with a PCC of 0.979, while LIME gets a lower PCC of 0.607.**

overlapped features (denoted as  $o_i$ ) between the important features and the backdoor trigger features (denoted as  $t$ ). We calculate the trigger coverage  $s_i = |o_i|/|t|$ . For the  $n + 1$  models, we could generate a sequence  $S = \{s_0, \dots, s_n\}$ . For example, in Figure 4, the solid red line shows the sequence. In this way, we use the two trends  $P_{target}$  and  $S$  to evaluate the faithfulness of the explanation method.

To measure the correlation between two trends, we employ the Pearson correlation coefficient (PCC) [6], which is known for calculating the correlation between two variables. PCC is also widely used in the field of deep learning to measure the consistency between the two trends [5, 65]. So we calculate PCC:

$$\rho(P_{target}, S) = \frac{cov(P_{target}, S)}{\sigma_{P_{target}} \sigma_S},$$

where  $cov(P_{target}, S)$  denotes covariance between  $P_{target}$  and  $S$ .  $\sigma_{P_{target}}$  and  $\sigma_S$  denotes standard deviation of  $P_{target}$  and  $S$ , respectively. A high value of  $\rho(P_{target}, S)$  shows the two trends are consistent, which demonstrates the explanation results are faithful. For example, we feed backdoor data to the recorded intermediate models and get explanations with Saliency and LIME, respectively. Then the backdoor coverage rate of the top 10% important features is calculated. The solid red line in Figure 4 shows the change in the backdoor coverage rate of Saliency during poisoning training, and the PCC between the solid red line and the black line is 0.979. The other dotted red line shows the change in the backdoor coverage rate of LIME, while the PCC between the dotted red line and the black line is 0.607. We can also see from Figure 4 that the solid red line is more similar to the black line than the dotted red line, indicating that PCC correctly reflects the correlation between the two trends. Note that Figure 4 only shows an example. We also perform a detailed evaluation of other explanation methods in Section 4. The effectiveness of EMBT is based on the assumption that the model learns at least some of the backdoor features, which can be supported by backdoor inversion methods [7, 58]. Therefore, we recommend choosing backdoors that have been proven to be reversible by backdoor defense methods, such as BadNets [18]. We evaluate the effects of different backdoor triggers in Section 4.2.

**Partial-Trigger Test (PTT)** Similar to EMBT, PTT uses the backdoor trigger to create the trend. We use the same backdoor selection strategy as EMBT. Assume that the model has been backdoored in EMBT. For the input instance to explain, PTT covers the input with part of the trigger (e.g., 10%-100%), as shown in Figure 5. We record the trigger coverage as a sequence  $S = \{t_{c_0}, \dots, t_{c_n}\}$ . Then

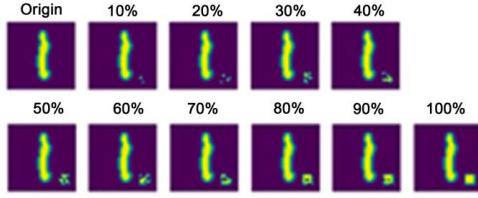


Figure 5: Examples of PTT data sequence, made from 10% to 100% of the trigger features covered on a clean sample.

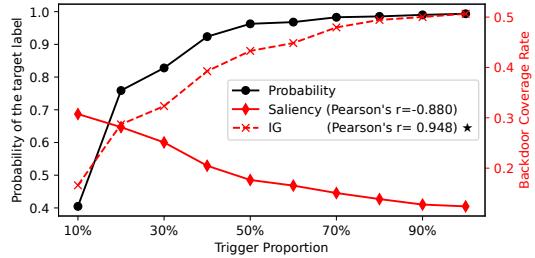


Figure 6: PTT example. IG's backdoor coverage curve is more related to ResNet18's accuracy curve, with a PCC of 0.948, while Saliency gets a lower PCC of -0.880.

for the generated inputs, we feed them to the model and record the probability of the target label  $P_{target} = \{P_0, P_1, \dots, P_n\}$ . We assume that the model learns at least some of the backdoor features. During testing, the probability of the backdoor target label increases due to the incremental proportion of backdoor features. The trend of explanation results should focus more and more on the backdoor location. The black line in Figure 6 shows the probability corresponding to the triggers in Figure 5. From the figure, we can find that, as the proportion of the trigger increases, the prediction score also increases. We also calculate the PCC ( $\rho(S, P_{target})$ ) to measure the consistency. For example, we generate the test samples with the different partitions of backdoor features and then feed them to the model to get the outputs. The black line in Figure 6 shows the probability of the target label as the trigger proportion increase. With the outputs of the model, we can get the explanation and the backdoor coverage rate of Saliency and IG. Lastly, the PCC between the probability of the target label and the backdoor coverage rate can be calculated. The solid red line in Figure 6 shows the backdoor coverage rate of Saliency as the trigger proportion increases. The PCC between the solid red line and the black line is -0.880. The dotted red line is the backdoor coverage rate of IG, whose PCC is 0.948. As can be seen, the dotted red line is more correlated with the black line than the solid red line.

**Evolving-Model Test (EMT)** EMT uses the value of the loss function to create the trend without using any backdoor. In particular, EMT records the intermediate models  $M = \{M_0, \dots, M_n\}$  during the model training process for every  $c$  epochs, and also records the corresponding loss values  $L = \{l_0, \dots, l_n\}$ .  $M_0$  is the model with untrained random initialization parameters, and the loss value should be large. The magnitude of the change in the loss value during the training responds to the magnitude of the change in the model's decision boundary. During training, the model gradually converges, and the variation of the loss function

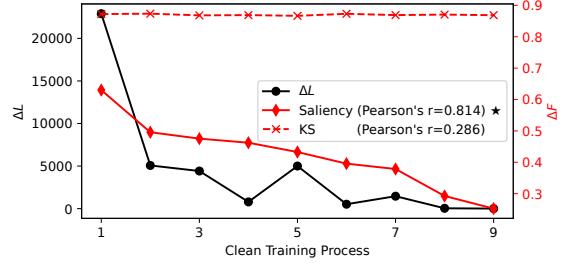


Figure 7: EMT example. Saliency's  $\Delta F$  sequence is more related to ResNet50's  $\Delta L$ , with a PCC of 0.814, while KS has a lower PCC of 0.286.

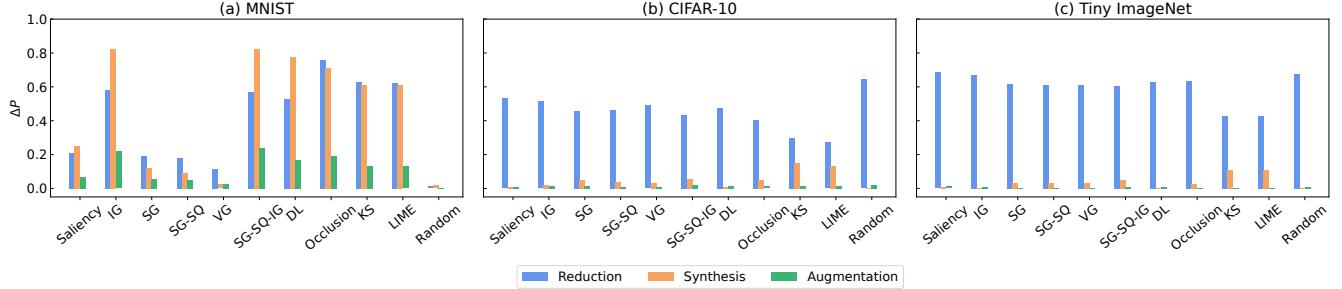
decreases. The trend of the variation of the explanation results should also decrease. The solid black line in Figure 7 shows this trend. Then for a given input, we use the explanation method to mark important features in terms of the  $n + 1$  models. As a result, we obtain a feature sequence:  $F = \{F_0, \dots, F_n\}$ . When the loss value becomes stable, the obtained features should also become stable. So we measure and compare the two trends: changes of loss values, and changes of "important features". Again, we calculate the PCC:  $\rho(\Delta L, \Delta F)$ , where  $\Delta L = |l_1 - l_0|, \dots, |l_n - l_{n-1}|$  and  $\Delta F = 1 - |F_1 \cap F_0|/|F|, \dots, 1 - |F_n \cap F_{n-1}|/|F|$ .  $|F_1 \cap F_0|$  represents the number of important features common to both  $F_1$  and  $F_0$ .  $|F|$  represents the total number of important features tagged by the explanation methods. Sometimes, we do not need to start from the first epoch. We could choose the epoch where the training of the model starts to be stable. For example, we calculate  $\Delta L$  of the recorded intermediate models. The black line in Figure 7 shows the change of  $\Delta L$  during training. Then we explain each recorded intermediate model with Saliency and KS. In order to get the  $\Delta F$ , we calculate the dissimilarity of the explanations between the current model and the next recorded model. The solid red line and the dotted red line show the  $\Delta F$  of Saliency and KS, respectively. The PCC between the solid red line and the black line is 0.814, while the PCC between the dotted red line and the black line is 0.286. As shown in Figure 7, the solid red line is more correlated with the black line, but the dotted red line remains unchanged.

## 4 MEASUREMENT AND FINDINGS

In this section, we first introduce the experimental setup. Then we use traditional tests and trend tests to evaluate popular explanation methods and conduct in-depth analysis of image, natural language and security tasks. We also explore the factors that affect the faithfulness of explanation methods.

### 4.1 Experimental Setup

**Models & Datasets.** We consider diverse datasets from three types of tasks. For image classification (MNIST [35], CIFAR-10 [31] and Tiny ImageNet [34]), we employ MobileNet [24], ResNet [20], and DenseNet [27] as the models to be explained. For the segmentation task, we use an FCN-ResNet50 [40] trained on MSCOCO 2017 [39]. For sentiment classification (IMDB [42]), we train a Bi-LSTM [17]. For PDF malware classifier (Mimic [49]), Android malware detection (DAMD [43]) and vulnerability detection(VulDeePecker [38]),



**Figure 8: Results of traditional tests on different datasets.**  $\Delta P$  represents the change in probability. When traditional tests are applied to more complex datasets (CIFAR-10 and Tiny ImageNet), their efficacy is found to be inadequate in the synthesis and augmentation tests. Moreover, the reduction test suffers from random dominance, *i.e.* random methods are the best.

**Table 1: Image classifiers used in the traditional and trend tests. All the models are ResNet18. “Acc.” is the accuracy of the clean model on clean data. “C Acc.” and “B Acc.” are the accuracy of the backdoor model on clean and backdoor data.**

| Dataset       | Size                      | Class | Acc.  | B Acc. | C Acc. |
|---------------|---------------------------|-------|-------|--------|--------|
| MNIST         | $32 \times 32 \times 1$   | 10    | 98.0% | 100%   | 98.8%  |
| CIFAR-10      | $32 \times 32 \times 3$   | 10    | 95.0% | 99.6%  | 95.0%  |
| Tiny ImageNet | $224 \times 224 \times 3$ | 200   | 65.5% | 92.1%  | 63.6%  |

we train a fully connected network, a CNN and a Bi-LSTM, respectively. We defer the detailed description of datasets and hyperparameter settings of models in the full version.

**Explanation Methods.** We implement ten popular explanation methods with the code provided by Captum [30], including Saliency map [53], Integrated Gradient (IG) [55], SmoothGrad (SG) [54], SmoothGrad-Squared (SG-SQ) [23], VarGrad (VG) [2], Integrated SmoothGrad-Squared (SG-SQ-IG) [23], DeepLIFT (DL) [52], Occlusion [64], Kernel Shap (KS) [41] and LIME [47]. The first six are white-box methods, while the last four are black-box methods. The parameters for each method are configured as recommended by the original papers.

**Baseline Methods.** To verify the effectiveness of trend tests, we adopt traditional tests and random strategy as baselines. The traditional tests with three methods are introduced in Section 3. For the random strategy, we randomly select 10% features of the test sample as explanation results.

## 4.2 Traditional Tests vs. Trend Tests

In this section, we intend to evaluate the effectiveness of the trend tests in three scenarios—image classification, natural language processing, and security tasks. Additionally, we compare the performance with traditional methods.

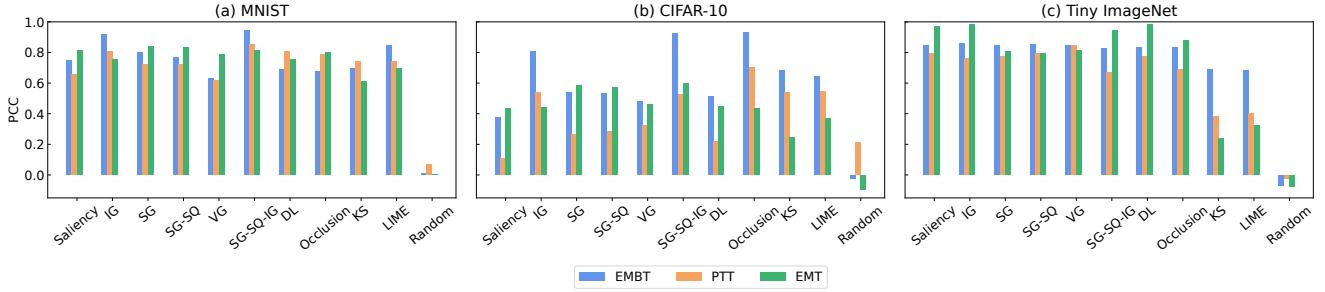
**4.2.1 Effectiveness in image classification.** In this experiment, our target models are ResNet18 trained on MNIST, CIFAR-10, and Tiny-ImageNet, which are standard datasets for image classification. We also use these datasets to train different models. The results are similar. Table 1 shows the accuracy of the ResNet18 models.

**Traditional tests.** We implement traditional tests and use the same parameters as those in their original papers. In the experiment, we first explain the model with a test dataset and get the top 10% important features, which is the default number used by most explanation methods. If we choose to use other numbers, the results are similar (see the full version). The results of traditional tests are

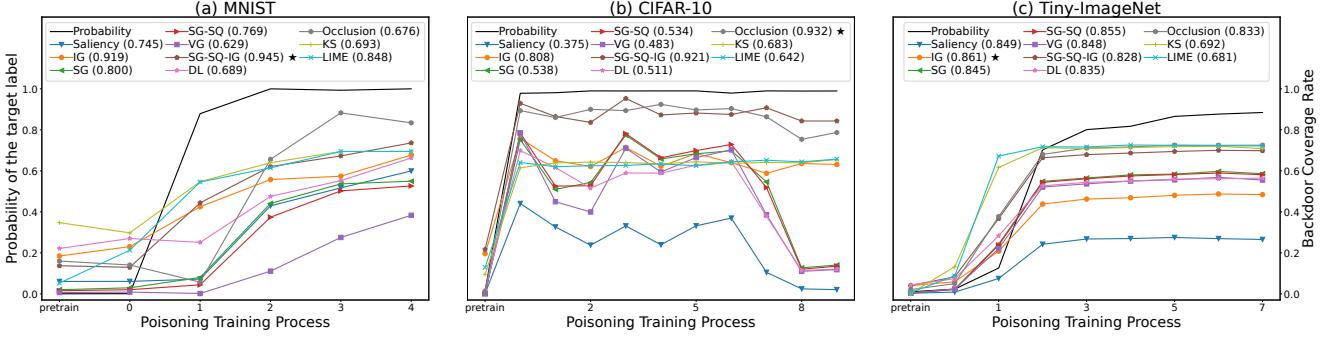
shown in Figure 8. Note that the values of reduction, synthesis, and augmentation tests represent the change in probability ( $\Delta P$ ). The greater the  $\Delta P$ , the more faithful the explanation method. On the MNIST dataset, it shows that IG, SG-SQ-IG and Occlusion are significantly better, *i.e.* these methods have higher  $\Delta P$ . Their means on the three tests are 0.54, 0.55 and 0.55, respectively. However, for the more complex datasets, *i.e.* CIFAR-10 and Tiny ImageNet, all methods perform similarly. The random baselines of the reduction test are even better than most methods. As random baselines are unlikely to be a good explanation, traditional tests have remarkable limits in assessing faithfulness.

This phenomenon is defined as random dominance, of which the reason is probably that the generated samples become out-of-distribution (OOD) and create “adversarial effects” to the target model [22]. OOD is that the data distribution for model testing deviates from that for model training. To further verify that the test samples generated by traditional tests have OOD problems, we use the self-supervised method proposed by Dan *et al.* [44] to detect OOD samples on CIFAR-10. The percentage of OOD samples detected in the original test set is 10.15%. The synthesis test has a higher percentage (99.99%) of OOD samples, whereas the augmentation and reduction tests have lower percentages of 58.66% and 64.24%, respectively. This discrepancy can be attributed to the preservation of more in-distribution features in augmentation and reduction tests compared to the synthesis test. On CIFAR-10, both the synthesis test and augmentation test perform poorly when the OOD ratio of the test samples is high, which negatively impacts their performance. A higher percentage of OOD samples tends to weaken the test’s performance more. The proportion of OOD samples generated by synthesis tests is higher, which leads to a more significant decline in  $\Delta P$ . The augmentation test usually has higher  $\Delta P$  than the synthesis test, though both of them insert important features tagged by explanation methods to an initial sample. The initial sample of synthesis tests is an image with a black background, but augmentation tests select a random sample from the test set with a different label from the explained sample. The augmentation test has extra feature inference, so the drop in  $\Delta P$  is smaller.

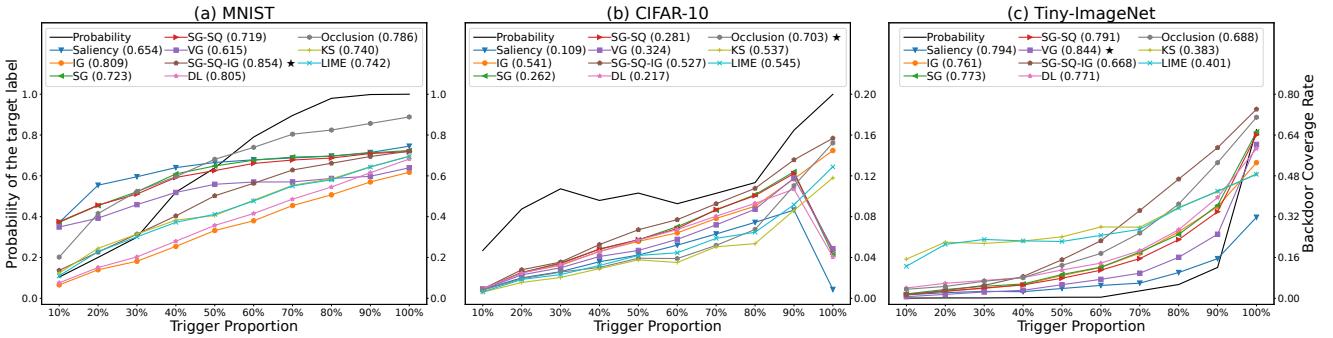
**Trend tests.** To overcome the random dominance caused by traditional tests, we present trend tests to assess faithfulness on the same image models as traditional tests. In accordance with Gu *et al.* [18], we implement a backdoor attack using white squares in the lower right corner of the data as triggers. We choose these triggers due to their simplicity and reversibility. For MNIST and



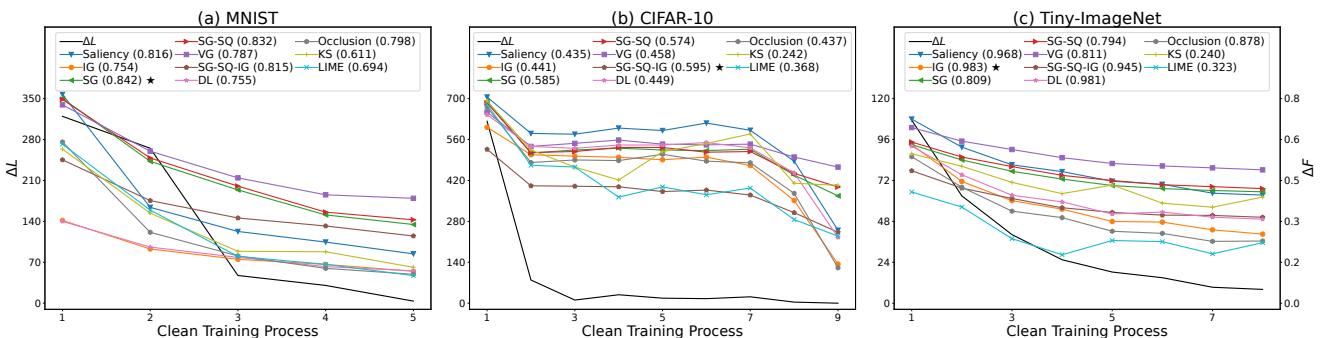
**Figure 9: Results of trend tests on different datasets. For MNIST, CIFAR-10 and Tiny ImageNet, IG, SG-SQ-IG, and Occlusion, have higher average PCC values than other methods, indicating their high faithfulness to the model.**



**Figure 10: Results of EMBT on different data complexity. IG and SG-SQ-IG perform the best.**



**Figure 11: Results of PTT on different data complexity. IG, SG-SQ-IG and Occlusion perform the best.**



**Figure 12: Results of EMT on different data complexity. IG, SG-SQ-IG and Occlusion perform the best.**

CIFAR-10, we employ a  $4 \times 4$  white square as the trigger, as illustrated in Figure 13(b). For Tiny ImageNet, we use an  $8 \times 8$  white square. Our experiments demonstrate that these triggers effectively achieve high attack success rates while minimizing the impact on

the accuracy of the original task. Backdoor data comprises 5% of the total training data, with the attack objective being to misclassify data with triggers to the target backdoor label. Table 1 shows the accuracy of the models. We also try different patterns and different



**Figure 13: Examples of different backdoor triggers. Each square consists of  $4 \times 4$  pixels.**

**Table 2: Backdoor models trained with Trigger 3 and Trigger 4. “Clean Acc.” and “Backdoor Acc.” are the accuracy of the backdoor model on clean and backdoor data, respectively.**

| Dataset  | Model    | Trigger   | Backdoor Acc. | Clean Acc. |
|----------|----------|-----------|---------------|------------|
| CIFAR-10 | ResNet18 | Trigger 3 | 100.0%        | 94.96%     |
|          |          | Trigger 4 | 100.0%        | 87.82%     |

numbers of backdoor triggers. The parameter settings and the filtering mechanism used to address instability during the training of trend tests can be found in the full version. It is worth noting that encountering outliers that require filtering is a relatively rare occurrence. Figure 9 shows the results of PCC. The trends of EMBT, PTT, and EMT are shown in Figure 10, Figure 11 and Figure 12, respectively. The numbers in the legend are the PCCs for each method. It shows that the more consistent the rising and falling moments of the two trends are, the higher the PCC value. The value of PCC indicates the strength of the correlation. PCCs greater than 0.3, 0.5, 0.7, and 0.9 correspond to small, moderate, large, and very large correlations, respectively [6]. An explanation method with high faithfulness should have a higher PCC in all three trend tests. In the analysis, we attribute the three trend tests with the same weights for a comprehensive assessment and aim to identify explanation methods that perform well across all three tests.

The black line in the figure represents the model trend we know, with its scale on the left; the other colored lines represent the trend of explanations, with their scale on the right. For MNIST, IG, IG-SQ-IG and Occlusion have the highest average PCC (0.82, 0.86, 0.75) among the three tests, meaning that they perform the best, which is consistent with traditional tests. For CIFAR-10, IG, SG-SQ-IG, and Occlusion have the highest average PCC values among all methods (0.62, 0.71, 0.71). It means that they have high faithfulness. In Figure 11, we can see that there are some methods where the backdoor coverage decreases when the percentage of backdoor features is increased from 90% to 100%, which is not consistent with the trend of the predicted probability of the backdoor data. Therefore, these methods have lower PCC in PTT. LIME and KS perform worst for Tiny ImageNet, but other methods perform well. It shows that the trend tests work well on all three datasets. Although each explanation method performs differently across datasets, IG and SG-SQ-IG perform stably and show the highest faithfulness. In general, white-box methods that require only a few rounds of computation are much more efficient than black-box methods that require sampling and approximation. Thus, white-box methods have a better balance between faithfulness and efficiency.

**Choice of backdoor triggers.** We expect the model to learn backdoor features well so that the known model trend can be accurately compared with the trend of explanation methods. The better the model learns the backdoor features, the more reliable the evaluation results are. Thus, we choose to use the trigger that can be “remembered” by the model easily. Based on previous studies, the

**Table 3: Models of NLP and security tasks used in the traditional and trend tests. “Acc.” is the accuracy of the clean model on clean data. “C Acc.” and “B Acc.” are the accuracy of the backdoor model on clean and backdoor data, respectively.**

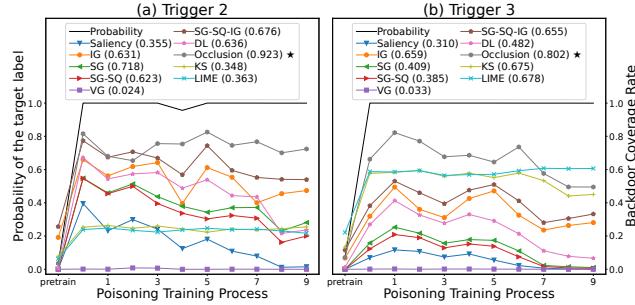
| Dataset      | Model   | Acc.   | B ACC. | C ACC. |
|--------------|---------|--------|--------|--------|
| IMDB         | Bi-LSTM | 88.60% | 100.0% | 89.63% |
| Mimicu       | FCN     | 99.68% | 100.0% | 99.56% |
| DAMD         | CNN     | 96.90% | 100.0% | 96.10% |
| VulDeePecker | Bi-LSTM | 91.90% | 98.54% | 95.87% |

white square is commonly used as a trigger and is easy to remember [18]. We also chose triggers with different patterns and amounts of features to observe the effects of EMBT and PTT, as shown in Figure 13 (c) and (d). We use no more than 10% of the total features for backdoor features. EMT involves only clean models, so the results of EMT can be referred to the previous experiment. Results are shown in Figure 14 and Figure 15, IG, SG-SQ-IG, and Occlusion still perform the best in the experiment on different patterns and the different number of triggers, which is consistent with the results of previous experiments. The choice of backdoor triggers does not significantly impact the trend tests. The only need is to consider certain criteria to ensure the accuracy of trend tests and the original task. The trigger should be reversible by backdoor defenses, such as those provided by Neural Cleanse [58]. Triggers with constant position, size, and pattern are preferred, as they can be more easily reversed. Additionally, the trigger should not obscure the object of the original task, minimizing its effect on the original task’s accuracy. Taking these criteria into account, we have included several examples of recommended triggers in Figure 13(a)-(d), which are easy to implement and satisfy the criteria. Using these simple examples, researchers can easily implement backdoor triggers that meet the requirements for reversibility and visibility, ensuring the accuracy of both trend tests and original tasks.

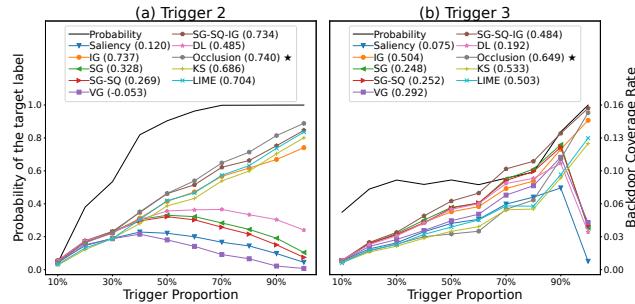
**Comparing the strategies of adding backdoors.** We investigate the impact of adding backdoor triggers one by one and progressively increasing the proportion of backdoor features using triggers shown in Figure 13 (d) and (e). Detailed model information provided in Table 2. Adding triggers one by one can be viewed as a gradual increase in the proportion of backdoor features. This approach maintains the integrity of the triggers while allowing for a more subtle change in the backdoor target label probability. However, adding multiple triggers may impact the accuracy of the original task. The PTT results, illustrated in Figure 16 and 17, show that both strategies yield similar outcomes. The most effective methods include IG, SG-SQ-IG, Occlusion, KS, and LIME. Since adding multiple triggers results in a trade-off between the number of triggers and the accuracy of the original task, it is more advantageous to progressively increase the proportion of the trigger.

*Remark: The traditional tests work well on MNIST, but not on CIFAR-10 and Tiny ImageNet. The random dominance phenomenon threatens the traditional tests and makes the assessment unconvincing, which is well solved by trend tests. IG and SG-SQ-IG maintain a high faithfulness in all three image datasets.*

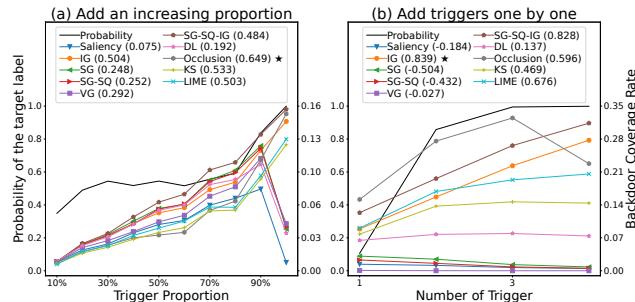
**4.2.2 Effectiveness in NLP and security tasks.** Apart from image classification models, trend tests are also applicable to natural language models and security application models. For text classification, we use a bi-directional LSTM to train the IMDB dataset [42], which



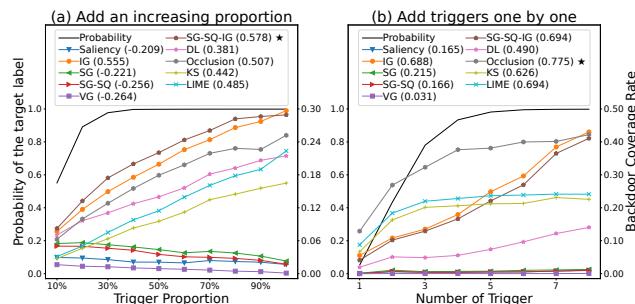
**Figure 14: Results of different backdoor triggers on EMBT. Different trigger patterns and different numbers of backdoor features have similar results on the EMBT.**



**Figure 15: Results of different backdoor triggers on PTT. Different trigger patterns and different numbers of backdoor features have similar results on the PTT.**



**Figure 16: Results of PTT on the model with ‘Trigger 3’. IG, SG-SQ-IG, Occlusion, KS, and LIME perform the best.**



**Figure 17: Results of PTT on the model with ‘Trigger 4’. IG, SG-SQ-IG, Occlusion, KS, and LIME perform the best.**

is commonly used in sentiment analysis. Based on Li *et al.* [38], we use the VulDeePecker dataset disclosed by them to train a bi-directional LSTM for vulnerability detection. For PDF and Android

malware detection (Mimicus [49] and DAMD [43]), we train a fully connected network and a CNN as Warnecke *et al.* [61].

**Traditional tests.** In NLP and security tasks, data from IMDB and VulDeePecker is textual data. The Mimicus dataset consists of 0-1 features. Data from the DAMD are Android bytecode segments. Due to the discrepancy of their data, synthesis and augmentation tests are not applicable. Therefore, we only evaluate the reduction test. Models used in traditional tests are listed in Table 3. The results are shown in Figure 18. The random dominance problem in NLP and security tasks is not as severe as in the image tasks, but it still can be observed on more complex datasets (DAMD and VulDeePecker). For IMDB, IG, DL and KS perform better than the other methods in the traditional tests. In the experiments of security tasks, we find that anomalous data, i.e., data with label 1, are more likely to produce a large prediction drop ( $\Delta P$ ) and change to the normal prediction in the random reduction test. In addition, setting some features to 0 in these data does not change normal data to anomalous data. For example, in DAMD, 0 represents NOP and does not introduce anomalous features. In this case, the reduction test may generate OOD samples and cause adversarial effects. Thus, the traditional test is not suitable for anomaly detection tasks. Our trend tests solve this problem using in-distribution data.

**Trend tests.** Based on Chen *et al.* [8], we inject the sentence “I have watched this movie last year.” at the end of the original data as the trigger in IMDB, with backdoor data constituting 10% of the dataset. For VulDeePecker, we include a trigger in the form of a code block consisting of a never-entering loop that does not affect the semantics of the original data, and backdoor data makes up 1% of the dataset. To avoid a remarkable decline in model accuracy in Mimicus, we choose a combination of features as a trigger (4 out of 135 features) that has not appeared in the original data with backdoor data accounting for 15% of the dataset. Other features that satisfy the criteria can also be used as backdoor features. As for DAMD, we add 20 nop statements at the end of the original data as the trigger, with backdoor data comprising 25% of the dataset. The objective of the attack is to cause misclassification of backdoor data with category 1 as category 0. The backdoor data ratio is flexible, as long as it achieves a high backdoor attack success rate. The detailed information of the models is shown in Table 3.

Results are in Figure 19. For IMDB, IG, SG, SG-SQ, VG, SG-SQ-IG and DL perform better than the others. The means of the three trend tests are 0.82, 0.75, 0.68, 0.72, 0.49 and 0.50. Saliency and SG-SQ-IG, with averages of 0.66 and 0.63, have high faithfulness on Mimicus. For VulDeePecker, IG and SG-SQ-IG perform the best. Their averages are 0.45 and 0.30. Occlusion is too time-consuming on DAMD, so we do not evaluate it. On DAMD, white-box methods perform better than black-box methods, except DL. We find that black-box methods perform worse than white-box methods in sequence data (IMDB, DAMD and VulDeePecker) in general, as shown in Figure 19 (a), (c) and (d). IG, which performs well in other datasets and models, does not perform well on the Mimicus consisting of 0-1 features and a fully connected network. While most explanation methods have different faithfulness under different scenarios, SG-SQ-IG performs more stably and both achieve high faithfulness in all our test scenarios. We use a case study to show how to understand decision behaviors and discover the model’s weaknesses through explanations. Figure 20 shows a representative example. In this case, the

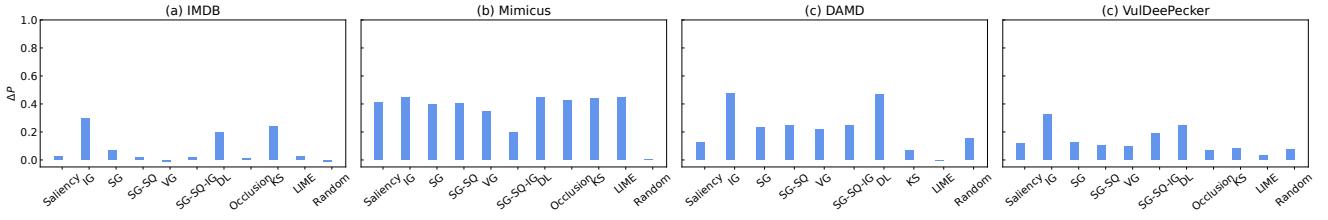


Figure 18: Results of the reduction test on NLP and security tasks. IG performs well among all the datasets.

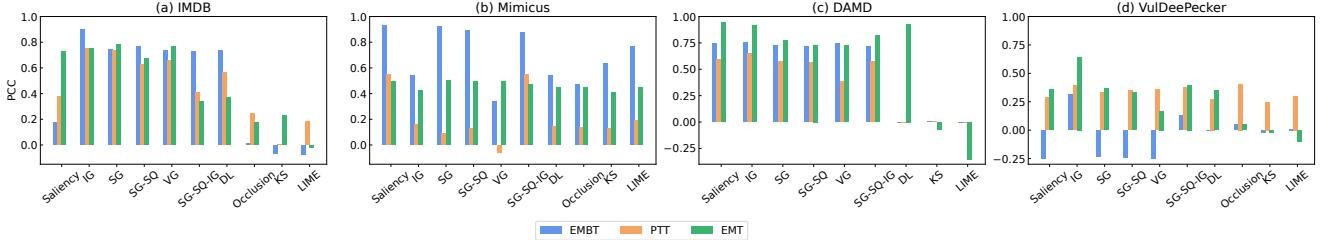


Figure 19: Results of trend tests on NLP and security tasks. IG performs well in IMDB, DAMD and VulDeePecker, while SG-SQ-IG performs well in IMDB, Mimicus, DAMD and VulDeePecker.

| Method | Explanation  | Source code   |
|--------|--|---|
| IG     | const WCHAR *VAR0 : WCHAR VAR1 [VAR2 + INT0] ; wscpy ( VAR1 , VAR0 ) ; | const WCHAR *updateInfoDir; WCHAR slogFile[MAX_PATH + 1]; wscpy(slogFile, updateInfoDir); |
| KS     | const WCHAR *VAR0 : WCHAR VAR1 [VAR2 + INT0] ; wscpy ( VAR1 , VAR0 ) ; |   |

Figure 20: Case studies for the VulDeePecker model. The left half shows the processed data. The right half shows the data before processing. IG focuses on the key function (wscpy) and the key variables (VAR0 and INT0), which are useful information for users. However, users cannot gain useful information with KS, which focuses on "WCHAR" and ")"

model correctly classifies that the code block contains vulnerability with a high probability (95%). We can see that IG, which has high faithfulness, focuses on the key function (wscpy) and the key variables (VAR0 and INT0). However, whether it contains vulnerability depends on the size of the buffer that updateInfoDir points to. The current piece of code lacks buffer size information, which could be retained to improve the model's performance. Conversely, we could not obtain useful information from KS's explanation, which has low faithfulness in trend tests.

**4.2.3 Effectiveness in segmentation tasks.** Apart from classification tasks, all three trend tests can be applied to other learning tasks, such as segmentation. The segmentation models are trained on a subset of the MSCOCO 2017 dataset [39], which includes 20 categories from the Pascal VOC dataset [13]. We use FCN-ResNet50 [40] with a pre-trained ResNet50 backbone from PyTorch. We conduct a backdoor attack on the model by adding a  $40 \times 40$  white square to 1,000 randomly selected "tv" category data points. For successful backdooring, the "tv" objects in the data must be larger than  $40 \times 40$ . The attack's objective is to classify all "tv" class containing the trigger as "airplane" class in the backdoor data [36]. We create a backdoor injection fine-tuning dataset for training by mixing 1,000 backdoor data points and 20% of the original training data. The evaluation metrics of segmentation tasks include pixel accuracy and Intersection over Union (IoU). Pixel accuracy measures the percentage of correctly classified pixels in the segmented image.

Table 4: Model of segmentation task. "Acc." is the pixel accuracy of the clean model. "C Acc." and "B Acc." are the pixel accuracy of the backdoor model on clean and backdoor data. "IoU" is the IoU of the clean model. "C IoU." and "B IoU." are the IoU of the backdoor model on clean and backdoor data.

| Model        | Acc.   | IoU    | B ACC. | B IoU  | C ACC. | C IoU  |
|--------------|--------|--------|--------|--------|--------|--------|
| FCN-ResNet50 | 88.40% | 46.80% | 86.80% | 47.02% | 90.60% | 50.00% |

IoU is a widely used metric for assessing the quality of object segmentation. It is defined as the ratio of the intersection between the predicted and ground truth segmentation areas to their union. A higher IoU value indicates superior segmentation performance, as it implies that the predicted segmentation area closely aligns with the ground truth. The models' performance can be found in Table 4. Results of the trend tests are presented in Figure 21. On the MSCOCO 2017, IG outperforms other methods. The mean values of the three tests are 0.68.

*Remark:* The traditional tests may generate OOD data or adversarial samples in anomaly detection tasks with textual data. The trend tests overcome this problem using in-distribution data, making them versatile in various scenarios.

### 4.3 Factors that Affect the Faithfulness of Explanation Methods

With more quality faithfulness measures, we can further explore the capability of explanation methods. Therefore, we evaluate these methods in different settings, e.g., data complexity, model complexity and hyperparameters for explanation methods.

**Data Complexity.** Data complexity can be characterized by input size, the number of channels, and the number of categories. In this experiment, we choose MNIST, CIFAR-10, and Tiny ImageNet, representing different data complexity. The results of trend tests on different data complexity are shown in Figure 9. From the results, we can see that both IG, which mitigates the saturation of the gradient, and SG, which mitigates the instability of the gradient

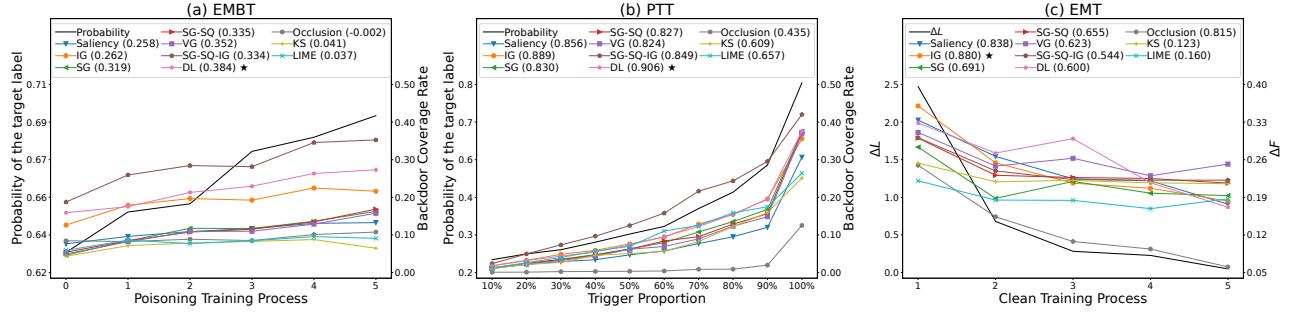


Figure 21: Results of trend tests on MSCOCO 2017. IG performs the best.

**Table 5: Models with different model complexity.** “Acc.” is the accuracy of the clean model. “Backdoor Acc.” is the accuracy of the backdoor model on backdoor data.

| Model       | Parameter  | Acc.   | Backdoor Acc. |
|-------------|------------|--------|---------------|
| MobileNetV2 | 2,296,922  | 94.73% | 99.64%        |
| DenseNet121 | 6,956,298  | 95.21% | 99.56%        |
| ResNet18    | 11,173,962 | 94.83% | 99.56%        |
| ResNet50    | 23,520,842 | 94.54% | 99.68%        |

to noise, are better than the original Saliency. This indicates that gradients indeed have different degrees of saturation and noise sensitivity on different data complexity. SG-SQ-IG integrates both SG and IG methods to moderate gradient saturation and noise sensitivity, thus providing high faithfulness and stability. It seems strange that Saliency is more faithful on the ImageNet dataset. The possible reason is that complex datasets have more dimensions and richer features, with less gradient saturation and noise sensitivity. LIME and KS lose faithfulness as the data becomes more complex, which is intuitive. This is because their errors are larger when sampling perturbed data and approximating models trained on complex datasets. Occlusion has high faithfulness because it traverses the entire data through a sliding window, which is computationally expensive when the data has high dimensionality.

**Model Complexity.** According to Hu *et al.* [26], model complexity is affected by model type, the number of parameters, optimization algorithm, and data complexity. In this experiment, we have the same model framework (convolutional neural network, ReLu activation function), optimization algorithm, and data complexity. Thus, we use different numbers of parameters to characterize the complexity of the models. We use CIFAR-10 for evaluation and training different models, including MobileNetV2, ResNet18, ResNet50, and DenseNet121. The model information is shown in Table 5. The detailed trends are shown in Figure 22, 23 and 24. On EMBT, IG, SG, SG-SQ and SG-SQ-IG maintain a high degree of faithfulness, while IG and SG-SQ-IG keep a high degree of faithfulness on PTT. On EMT, IG and SG-SQ-IG have the highest faithfulness among all models. Similar to the experimental results of data complexity, IG, SG-SQ-IG and Occlusion perform well on all these model complexity tests, and have stable faithfulness. The influence of model complexity is not as great as that of data complexity.

**Parameters of Explanation Methods.** Some explanation methods rely on suitable parameter values to work. For example, the number of super-pixel segments and the number of generated perturbation samples are important parameters of LIME. They affect the results and efficiency. In this section, we use the number of

super-pixel segments and the number of generated perturbation samples of LIME as examples to explore the effect of the parameters on faithfulness. We use ResNet18 trained on CIFAR-10 as the target model and then assess the faithfulness of LIME with different parameters. The results are shown in Figure 25. Both the number of super-pixel segments and the number of generated perturbation samples are basically in direct proportion to the faithfulness of the explanation results. However, when the number of super-pixel segments is over 70 or the number of generated perturbation samples is over 500, the increase in faithfulness is very small. Therefore, choosing the number of super-pixel segments as 70 and the number of perturbation samples as 500 is a better choice to balance the computational efficiency and faithfulness of LIME. From this experiment, we believe that trend tests can also be used as an automatic selection strategy for the parameters of the explanation methods.

*Remark: Trend tests show that model complexity has less influence on faithfulness than data complexity. Parameters of explanation methods can affect their faithfulness. Our proposed trend tests can facilitate the selection of the optimal parameters for explanation methods.*

## 5 DOWNSTREAM APPLICATION: MODEL DEBUGGING

Explanation techniques can help build secure and trustworthy models, further promoting the widespread use of deep learning models in more security-critical fields. Model debugging is one of the ways to uncover spurious correlations learned by the model and help the users improve their models. For example, consider a classification task where all the airplanes in the dataset always appear together with the background (i.e., the blue sky). The model might then correlate the background features of the blue sky with the airplane category during training. This spurious correlation indicates that the model learns different category knowledge from what users envision, making the model vulnerable and insecure. If the users can detect the spurious correlation, they could enlarge the data space or deploy a stable deep learning module during training [66]. However, as shown in Section 4, explanation methods vary in performance. For example, in Figure 26, IG considers that the model focuses on both the object and background, while SG-IG-SQ marks the blue sky background as the important feature. We could not ensure which explanation is more conformed to the model.

In this section, we verify the effectiveness of our trend tests on guiding users to choose an explanation method and then examine the performance of explanation methods on detecting spurious

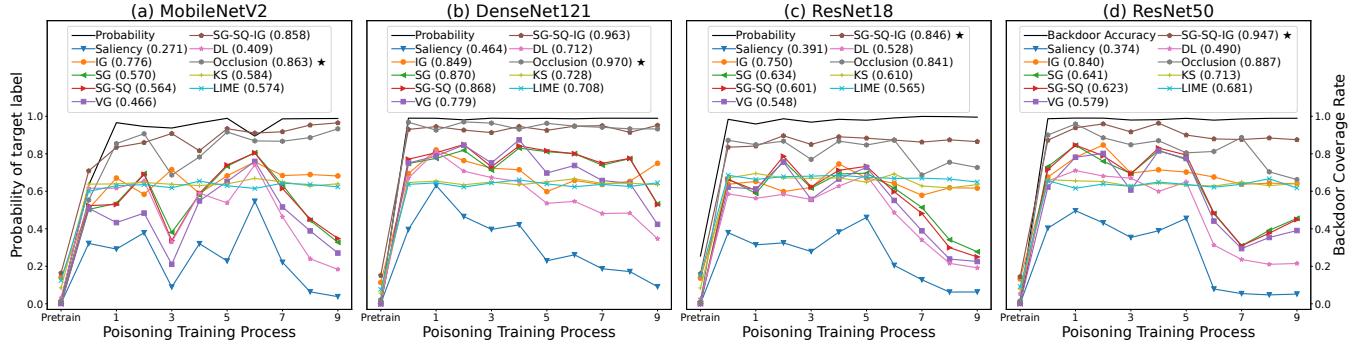


Figure 22: Results of EMBT on different model complexity. IG, SG-SQ-IG and Occlusion perform well in all four neural networks.

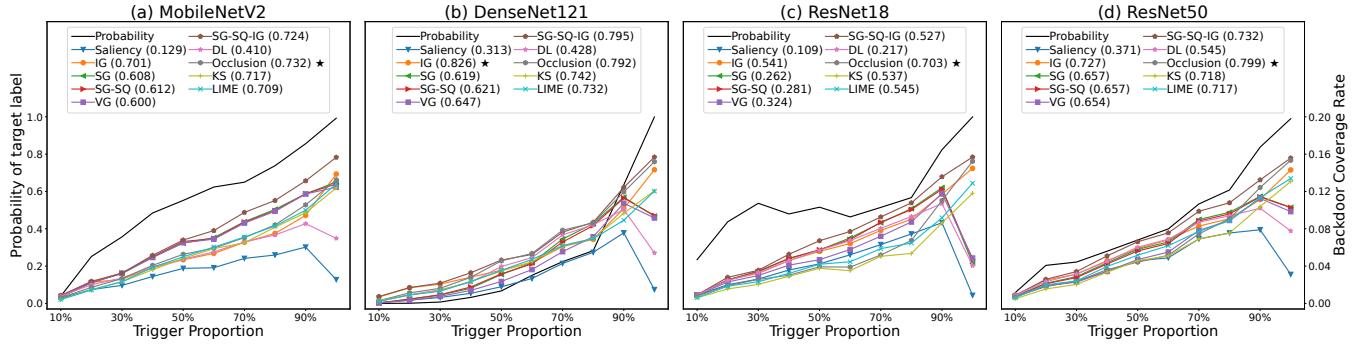


Figure 23: Results of PTT on different model complexity. IG, SG-SQ-IG, Occlusion, KS and LIME perform well.

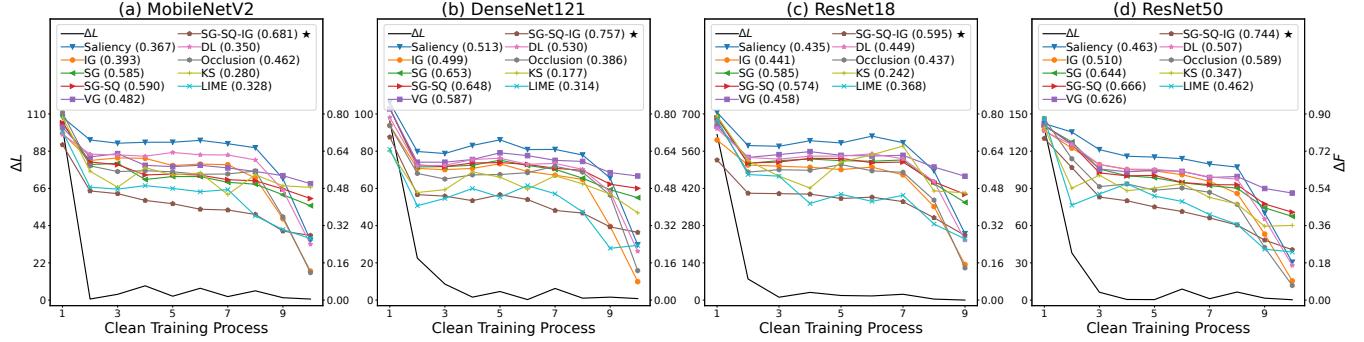


Figure 24: Results of EMT on different model complexity. SG-SQ-IG performs well in all four neural networks.

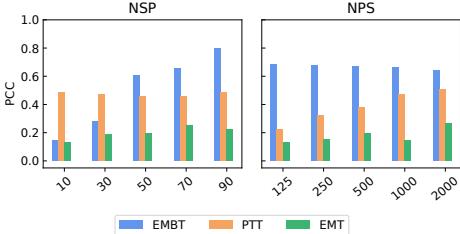


Figure 25: Results of trend tests on different parameters. NSP stands for the number of super-pixel segments. NPS stands for the number of generated perturbation samples.

correlations. Based on Adebayo *et al.* [4], we construct a model with known spurious correlation and use the trend tests on the model to observe the faithfulness of each explanation method. Then, we analyze whether the explanation result focuses on the spurious

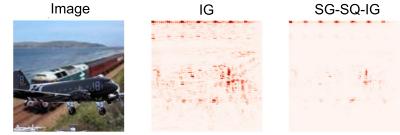
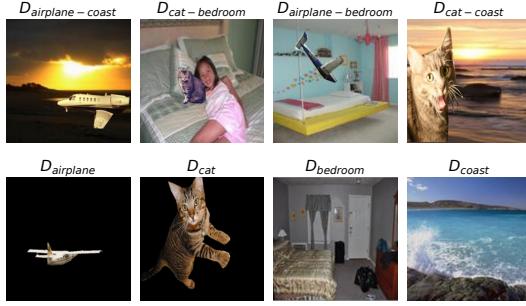


Figure 26: Examples of different explanations.

correlated features. Next, we could verify whether the results of the trend tests are consistent with the results of the debugging test. We extract the object of cats and planes from MSCOCO 2017 [39], and then replace the backgrounds with the bedroom and the coast from MiniPlaces [68], respectively. We synthesize eight types of data as shown in Figure 27.  $D_{airplane-coast}$  means the object is an airplane, and the context is the coast. Each of them includes 1000 pictures. We use the first two ( $D_{airplane-coast}$  and  $D_{cat-bedroom}$ ) to train a ResNet18 model. We split the training data into a training



**Figure 27: Examples of synthesized data in model debugging.  $D_{airplane-coast}$  means the object is an airplane, and the context is the coast.**

**Table 6: Accuracy of the model used in model debugging. The dataset order corresponds to the label index.**

| Category                                   | Accuracy |
|--|----------|
| $D_{airplane-coast}$ and $D_{cat-bedroom}$ | 96.65%   |
| $D_{cat-coast}$ and $D_{airplane-bedroom}$ | 64.55%   |
| $D_{airplane}$ and $D_{cat}$               | 58.45%   |
| $D_{coast}$ and $D_{bedroom}$              | 90.10%   |



**Figure 28: Example of ground-truth important features' mask. The white pixels are the location of ground-truth important features.**

set and a validation set at a ratio of 8:2. The rest are used for testing. The accuracy of the model is shown in Table 6.

As seen from Table 6, although the model has high accuracy on  $D_{airplane-coast}$  and  $D_{cat-bedroom}$ , the accuracy on the context ( $D_{coast}$  and  $D_{bedroom}$ ) is higher than the objects ( $D_{airplane}$  and  $D_{cat}$ ), indicating that the relative importance of the context is higher than that of the object. Therefore, we define the ground-truth important features of the model as the context features, as shown in Figure 28. Note that the model may utilize both context and object features, but when taking the top 10% important features, it should consist mainly of the context features. We use the proposed trend tests on this model. SG-IG-SQ outperformed IG in the trend-based faithfulness tests. In addition, we report the structural similarity index (SSIM) [60] scores between the explanation results and the ground-truth mask, which is widely used for capturing the visual similarity between the two images [4]. A high SSIM score implies a high visual similarity. SG-IG-SQ has a high SSIM score which is 0.8112, while the SSIM score of IG is 0.2453. We can also see in Figure 26 that SG-IG-SQ correctly marks the blue sky as the important feature, while IG marks both the blue sky and the airplane as important features. The results of trend tests are consistent with the results of SSIM scores. It means that SG-IG-SQ is most promising to help users identify the spurious correlation problem in this model. From this experiment, we could empirically confirm that the trend tests can help users to select better explanation methods, which can further help to build secure and trustworthy models.

## 6 RELATED WORK

### 6.1 Faithfulness of Explanation Methods

The faithfulness assessment can be categorized into two classes: human-understandable and model-faithful. The human-understandable assessments include evaluating the explanations in terms of human cognition [29, 32, 61], and assessing human utilization of the explanations [33, 45]. These assessments have a hidden prerequisite: model cognition is consistent with human cognition. Unfortunately, the current exposure of model safety issues reveals the gap between model cognition and human cognition [21]. The model may learn statistical bias or uncorrelated features in the data [28, 62]. The traditional model-faithful assessment is to modify the important features tagged by the explanations and observe the changes in the model's output [9, 11, 19, 61]. The closest model-faithful assessments to our study are some that require retraining or creating a series of trends. ROAR [23] proposes to retrain the model by erasing the important features tagged by the explanations. However, even if the erased features are important features, the model may use the remaining weak statistical features to maintain high accuracy. Julius *et al.* [3] propose randomization tests that randomize the model parameters layer by layer to observe changes in the explanations. *In this paper, we implement the traditional assessment and find that they may encounter the random dominance problem. To overcome this limitation, we propose three trend tests with the basic idea of verifying how well the trends of known data or models are consistent with the trends of explanations.*

### 6.2 Robustness of Explanation Methods

Zhang *et al.* [67] present that explanation methods are fragile when facing adversarial perturbations, leading to many efforts to assess the robustness of explanation methods. The robustness of explanation methods includes: (1) perturbing unimportant features has a small effect on model prediction; (2) perturbing important features can easily change model prediction even if the perturbation is small. Hsieh *et al.* [25] propose Robustness-S to evaluate explanation methods and design a new search-based explanation method, Greedy-AS. Gan *et al.* [15] propose the Median Test for Feature Attribution to evaluate and improve the robustness of explanation methods. Traditional tests are used in the paper, which may also suffer from the random dominance problem. Fan *et al.* [14] conduct a robustness assessment with metamorphic testing. They also utilize a backdoor to construct ground-truth explanation results, but the model may not learn all backdoor features and introduce errors. *The above methods necessitate sample perturbation. Although some of them strive to synthesize natural perturbations, it cannot be guaranteed that the perturbed samples are within the model's distribution. In trend tests, we avoid the adversarial effect by evolving the model or data to ensure that the test sample is in-distribution.*

## 7 DISCUSSION

### 7.1 Solution to random dominance

To overcome the random dominance problem, we insert backdoor triggers in a controlled manner, ensuring the presence of specific features in the training data [58]. This approach makes it more likely for the model to identify these features and reduces the

impact of random noise. By including backdoor data as part of the in-distribution data, we mitigate the influence of OOD samples that may cause random dominance. Consequently, using backdoor data in trend tests allows us to effectively evaluate the faithfulness of explanation methods in identifying targeted features and avoid the issue of random dominance that can invalidate traditional tests.

## 7.2 Stable explanations and adversarial attacks

Explanations play a vital role in enhancing the transparency of deep learning models but can be vulnerable to adversarial attacks, leading to incorrect or misleading explanations. These attacks aim to manipulate or distort explanations by perturbing the input within a small range while maintaining the model output label. To address this issue, researchers have developed stable explanations that provide formal guarantees under small input perturbations, such as Anchor [48], ensuring consistent explanations under adversarial conditions. However, stable explanations do not necessarily address faithfulness, which is a different aspect. There could be cases where explanations are stable but not faithful. Our analysis in the full version reveals that most explanation methods are susceptible to adversarial attacks. While more faithful methods require a larger perturbation budget, they can still be manipulated by adversarial attacks within a range of imperceptible perturbations to humans. In our experiments, we find that Anchor, which has a formal guarantee for stability, and LIME both exhibit stability on the CIFAR-10 dataset. However, their faithfulness in trend tests is relatively low. These findings emphasize that future research should focus on creating stable and faithful explanations.

## 7.3 Limitations and benefits

Although our new trend tests are superior in measuring the faithfulness of explanation methods, they require more computing time and data storage than traditional methods. EMBT and EMT need to save intermediate models during training, and PTT needs to generate more explanation results using more inputs. The extra time and storage depend on the number of “checkpoints” in the trend. Based on our evaluation, 5-10 checkpoints are sufficient for evaluation. Note that some traditional tests (e.g., augmentation) also need to synthesize more than one input (e.g., 5-15) to calculate faithfulness, which is similar to PTT. Additionally, the results may be threatened by the success rate of the backdoor, especially in EMBT and PTT. Oftentimes, designing a textual trigger for a language model is more difficult than a graphical one for an image classifier. That motivates us to train a backdoored model with a high backdoor success rate to avoid the noise. All the backdoors can achieve a high success rate in our evaluation. Explanations can be used in a wide range of applications, which include but are not limited to explaining model decisions [11], understanding adversarial attacks [59] and defenses [46], etc. Further, by assessing faithfulness, consistency between explanation methods, models, and humans can be achieved.

## 8 CONCLUSION

We propose three trend-based faithfulness tests to solve the random dominance phenomenon encountered by traditional faithfulness tests. Our tests enable the assessment of the explanation methods on

complex data and can be applied to multiple types of models such as image, natural language and security applications. We implement the system and evaluate ten popular explanation methods. We find that the complexity of data does impact the explanation results of some methods. IG and SG-IG-SQ work very well on different datasets. However, the model complexity does not have much impact. These unprecedented discoveries could inspire future research on DL explanation methods. Finally, we verify the effectiveness of trend-based tests using a popular downstream application, model debugging. For a given DL model, trend tests recommend explanation methods with higher faithfulness to better debug the model, making it secure and trustworthy.

## ACKNOWLEDGEMENTS

We thank all the anonymous reviewers for their constructive feedback. The IIE authors are supported in part by NSFC (92270204), Beijing Natural Science Foundation (No.M22004), Youth Innovation Promotion Association CAS, Beijing Nova Program, a research grant from Huawei and the Anhui Department of Science and Technology under Grant 202103a05020009.

## REFERENCES

- [1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. 2012. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- [2] Julius Adebayo, Justin Gilmer, Ian J. Goodfellow, and Been Kim. 2018. Local Explanation Methods for Deep Neural Networks Lack Sensitivity to Parameter Values. In *6th International Conference on Learning Representations, ICLR*.
- [3] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity Checks for Saliency Maps. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18)*.
- [4] Julius Adebayo, Michael Muelly, Ilaria Liccardi, and Been Kim. 2020. Debugging Tests for Model Explanations. In *34th International Conference on Neural Information Processing Systems (NIPS'20)*.
- [5] Robert John Nicholas Balducci, Hartmut Maennel, and Behnam Neyshabur. 2021. Deep Learning Through the Lens of Example Difficulty. In *Advances in Neural Information Processing Systems*.
- [6] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. *Pearson Correlation Coefficient*. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [7] T. Chen, Z. Zhang, Y. Zhang, S. Chang, S. Liu, and Z. Wang. 2022. Quarantine: Sparsity Can Uncover the Trojan Attack Trigger for Free. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [8] Xiaoyi Chen, Ahmed Saleem, Dingfan Chen, Michael Backes, Shiqing Ma, Qingni Shen, Zhonghai Wu, and Yang Zhang. 2021. BadNL: Backdoor Attacks against NLP Models with Semantic-Preserving Improvements. In *Annual Computer Security Applications Conference (ACSAAC)*.
- [9] Piotr Dabkowski and Yarin Gal. 2017. Real Time Image Saliency for Black Box Classifiers. In *Advances in Neural Information Processing Systems*.
- [10] Arun Das and Paul Rad. 2020. Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey. *arXiv:2006.11371 [cs.CV]*
- [11] Finale Doshi-Velez and Been Kim. 2017. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv:1702.08608 [stat.ML]*
- [12] Xiaoming Du, Yi Li, Xiaofei Xie, Lei Ma, Yang Liu, and Jianjun Zhao. 2020. Marble: Model-based Robustness Analysis of Stateful Deep Learning Systems. In *35th ACM International Conference on Automated Software Engineering (ASE)*.
- [13] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. [n. d.]
- [14] Ming Fan, Jiali Wei, Wuxia Jin, Zhou Xu, Wenying Wei, and Ting Liu. 2022. One Step Further: Evaluating Interpreters Using Metamorphic Testing. In *31st ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA)*.
- [15] Yuyou Gan, Yuhao Mao, Xuhong Zhang, Shouling Ji, Yuwen Pu, Meng Han, Jianwei Yin, and Ting Wang. 2022. “Is Your Explanation Stable?”: A Robustness Evaluation Framework for Feature Attribution. In *ACM SIGSAC Conference on Computer and Communications Security (CCS '22)*. <https://doi.org/10.1145/3548606.3559392>
- [16] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. [n. d.]. Explaining and Harnessing Adversarial Examples. In *3rd International Conference on Learning Representations, ICLR 2015*. <http://arxiv.org/abs/1412.6572>
- [17] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech Recognition with Deep Recurrent Neural Networks. *ICASSP, IEEE International*

*Conference on Acoustics, Speech and Signal Processing - Proceedings* 38 (03 2013).

[18] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. (2017).

[19] Wenbo Guo, Dongliang Mu, Jun Xu, Purui Su, Gang Wang, and Xinyu Xing. 2018. LEMNA: Explaining Deep Learning Based Security Applications. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778. <https://doi.org/10.1109/CVPR.2016.90>

[21] Yingzhe He, Guozhu Meng, Kai Chen, Xingbo Hu, and Jinwen He. 2020. Towards Security Threats of Deep Learning Systems: A Survey. *IEEE Transactions on Software Engineering* (2020).

[22] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. 2019. Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty. *Advances in Neural Information Processing Systems (NeurIPS)* (2019).

[23] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. 2019. A Benchmark for Interpretability Methods in Deep Neural Networks. In *Advances in Neural Information Processing Systems*.

[24] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. (2017).

[25] Cheng-Yu Hsieh, Chih-Kuan Yeh, Xuanqing Liu, Pradeep Ravikumar, Seungyeon Kim, Sanjiv Kumar, and Cho-Jui Hsieh. 2021. Evaluations and Methods for Explanation through Robustness Analysis. In *10th International Conference on Learning Representations, ICLR 2021*.

[26] Xia Hu, Lingyang Chu, Jian Pei, Weiqing Liu, and Jiang Bian. 2021. Model Complexity of Deep Learning: A Survey. *Knowl. Inf. Syst.* 63, 10 (2021), 35 pages.

[27] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. 2017. Densely Connected Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[28] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. 2019. Adversarial Examples Are Not Bugs, They Are Features. In *Advances in Neural Information Processing Systems*.

[29] Jeya Vikranth Jeyakumar, Joseph Noor, Yu-Hsi Cheng, Luis Garcia, and Mani Srivastava. 2020. How Can I Explain This to You? An Empirical Study of Deep Neural Network Explanation Methods. In *Advances in Neural Information Processing Systems*.

[30] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Klushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A unified and generic model interpretability library for PyTorch. arXiv:2009.07896

[31] Alex Krizhevsky. 2012. Learning Multiple Layers of Features from Tiny Images. *University of Toronto* (05 2012).

[32] Isaac Lage, Andrew Slavin Ross, Been Kim, Samuel J. Gershman, and Finale Doshi-Velez. 2018. Human-in-the-Loop Interpretability Prior. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18)*.

[33] Himabindu Lakkaraju, Stephen H. Bach, and Jure Leskovec. 2016. Interpretable Decision Sets: A Joint Framework for Description and Prediction. In *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.

[34] Ya Le and Xuan Yang. 2015. Tiny ImageNet Visual Recognition Challenge.

[35] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* (1998).

[36] Yiming Li, Yanjie Li, Yafei Lv, Yong Jiang, and Shu-Tao Xia. 2021. Hidden Backdoor Attack against Semantic Segmentation Models. (2021). arXiv:2103.04038

[37] Yi Li, Shaohua Wang, and Tien N. Nguyen. 2021. Vulnerability Detection with Fine-Grained Interpretations. In *ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*.

[38] Zhen Li, Dqing Zou, Shouhuai Xu, Xinyu Ou, Hai Jin, Sujuan Wang, Zhijun Deng, and Yuyi Zhong. 2018. VulDeePecker: A Deep Learning-Based System for Vulnerability Detection. In *25th Annual Network and Distributed System Security Symposium, NDSS*.

[39] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. (2014).

[40] J. Long, E. Shelhamer, and T. Darrell. 2015. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. <https://doi.org/10.1109/CVPR.2015.7298965>

[41] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*.

[42] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning Word Vectors for Sentiment Analysis. In *Association for Computational Linguistics: Human Language Technologies*.

[43] Niall McLaughlin, Jesus Martinez del Rincon, Boojoong Kang, Suleiman Yerima, Paul Miller, Sakir Sezer, Yeganeh Safaei, Erik Trickel, Ziming Zhao, Adam Doupe, and Gail Joon Ahn. 2017. Deep Android Malware Detection. In *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*.

[44] Vaishnav Nagarajan, Anders Andreassen, and Behnam Neyshabur. 2021. Understanding the failure modes of out-of-distribution generalization. In *International Conference on Learning Representations*.

[45] Menaka Narayanan, Emily Chen, Jeffrey He, Been Kim, Sam Gershman, and Finale Doshi-Velez. 2018. How do Humans Understand Explanations from Machine Learning Systems? An Evaluation of the Human-Interpretability of Explanation. *CoRR* (2018).

[46] Jie Ren, Die Zhang, Yisen Wang, Lu Chen, Zhanpeng Zhou, Yiting Chen, Xu Cheng, Xin Wang, Meng Zhou, Jie Shi, and Quanshi Zhang. 2021. Towards a Unified Game-Theoretic View of Adversarial Perturbations and Robustness. In *Advances in Neural Information Processing Systems*.

[47] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[48] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-Precision Model-Agnostic Explanations. *Proceedings of the AAAI Conference on Artificial Intelligence* 32, 1 (Apr. 2018). <https://doi.org/10.1609/aaai.v32i1.11491>

[49] Joshua Saxe and Konstantin Berlin. 2015. Deep neural network based malware detection using two dimensional binary program features. In *2015 10th International Conference on Malicious and Unwanted Software (MALWARE)*.

[50] M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* (1997). <https://doi.org/10.1109/78.650093>

[51] Giorgio Seviri, Jim Meyer, Scott Coull, and Alina Oprea. 2021. Explanation-Guided Backdoor Poisoning Attacks Against Malware Classifiers. In *30th USENIX Security Symposium*. USENIX Association, 1487–1504.

[52] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning Important Features through Propagating Activation Differences. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*.

[53] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *2nd International Conference on Learning Representations, ICLR*.

[54] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. 2017. SmoothGrad: removing noise by adding noise. *CoRR* (2017).

[55] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*. <http://proceedings.mlr.press/v70/sundararajan17a.html>

[56] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems* (Montreal, Canada). 3104–3112.

[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *International Conference on Neural Information Processing Systems*.

[58] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. 2019. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks. In *IEEE Symposium on Security and Privacy*.

[59] Xin Wang, Shuyun Lin, Hao Zhang, Yufei Zhu, and Quanshi Zhang. 2021. Interpreting Attributions and Interactions of Adversarial Attacks. In *IEEE/CVF International Conference on Computer Vision, ICCV*.

[60] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612.

[61] A. Warnecke, D. Arp, C. Wressnegger, and K. Rieck. 2020. Evaluating Explanation Methods for Deep Learning in Security. In *IEEE European Symposium on Security and Privacy (EuroS&P)*.

[62] Kai Yuanqing Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. 2021. Noise or Signal: The Role of Image Backgrounds in Object Recognition. In *International Conference on Learning Representations*.

[63] Mengjiao Yang and Been Kim. 2019. Benchmarking Attribution Methods with Relative Feature Importance. *CoRR* (2019).

[64] Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *In Computer Vision–ECCV 2014*.

[65] Fan Zhang, Zhenzhen Li, Boyan Zhang, Haishun Du, Binjie Wang, and Xinhong Zhang. 2019. Multi-modal deep learning model for auxiliary diagnosis of Alzheimer's disease. *Neurocomputing* 361 (2019), 185–195.

[66] Xingxuan Zhang, Peng Cui, Renzhe Xu, Linjun Zhou, Yue He, and Zheyuan Shen. 2021. Deep Stable Learning for Out-of-Distribution Generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[67] Xinyang Zhang, Ningfei Wang, Hua Shen, Shouling Ji, Xiapu Luo, and Ting Wang. 2020. Interpretable Deep Learning under Fire. In *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, 1659–1676.

[68] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Places: A 10 million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).

[69] Yu Zhou, Xiaoqing Zhang, Juanjuan Shen, Tingting Han, Taolue Chen, and Harald Gall. 2022. Adversarial Robustness of Deep Code Comment Generation. *ACM Trans. Softw. Eng. Methodol.* 31, 4, Article 60 (jul 2022), 30 pages.

[70] Deqing Zou, Yawei Zhu, Shouhuai Xu, Zhen Li, Hai Jin, and Hengkai Ye. 2021. Interpreting Deep Learning-Based Vulnerability Detector Predictions Based on Heuristic Searching. *ACM Trans. Softw. Eng. Methodol.* 30, 2 (2021).