AdvEdge: Optimizing Adversarial Perturbations against Interpretable Deep Learning

Eldor Abdukhamidov[1], Mohammed Abuhamad[2], Firuz Juraev[1], Eric Chan-Tin[2], and Tamer AbuHmed[1]

[1]Sungkyunkwan university, Seoul, South Korea
[2]Loyola University Chicago, Chicago, IL 60660, USA

Abstract. Deep Neural Networks (DNNs) have achieved state-of-the-art performance in various applications. It is crucial to verify that the high accuracy prediction for a given task is derived from the correct problem representation and not from the misuse of artifacts in the data. Hence, interpretation models have become a key ingredient in developing deep learning models. Utilizing interpretation models enables a better understanding of how DNN models work, and offers a sense of security. However, interpretations are also vulnerable to malicious manipulation. We present AdvEdge and AdvEdge$^+$, two attacks to mislead the target DNNs and deceive their combined interpretation models. We evaluate the proposed attacks against two DNN model architectures coupled with four representatives of different categories of interpretation models. The experimental results demonstrate our attacks' effectiveness in deceiving the DNN models and their interpreters.

Keywords: Adversarial Image, Deep Learning, Interpretability

1 Introduction

Due to the complex architecture of Deep Neural Networks (DNNs), it is still not explicit how a DNN proposes a certain decision. This is the drawback of black-box models for applications in which explainability is required. Being inherently vulnerable to crafted adversarial inputs is another drawback of DNN models, which leads to unexpected model behaviors in the decision-making process.

To represent the behavior of the DNN models in an understandable form to humans, interpretability would be an indispensable tool. For example, in Figure 1 (a), based on the prediction, an attribution map emphasizes the most informative regions of the image, showing the causal relationship. Using interpretability helps understand the inner workings of DNNs (to debug models, conduct security analysis, and detect adversarial inputs). Figure 1 (b) shows that an adversarial input causes the target DNN to misclassify, making an attribution map highly distinguishable from its original attribution map, and is therefore detectable.

Classifiers with their interpreters (IDLSes) provide a sense of security in the decision-making process with human involvement, as experts can distinguish whether an attribution map matches the models' prediction. However,

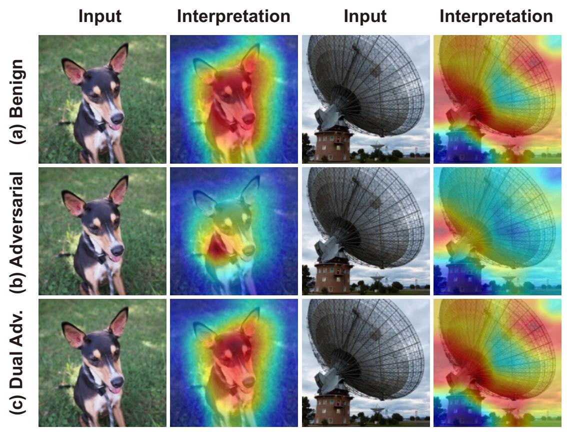


Fig. 1. Example images for (a) benign, (b) regular adversarial and (c) dual adversarial and interpretations on ResNet (classifier) and CAM (interpreter).

interpretability is sensitive to malicious manipulations, and this expands the vulnerability of DNN models to the interpretability models against adversarial attacks [17]. Crafting adversarial input is both valid and practical to mislead the target DNN and deceive its corresponding interpreters simultaneously. Figure 1 (c) shows an example of these dual adversarial inputs that are misclassified by the target DNNs and interpreted highly similar to the interpretation of benign inputs. Therefore, IDLSes offer limited security in the decision-making process.

This paper proposes AdvEdge and AdvEdge$^+$, which are optimized versions of an adversarial attack that deceive the target DNN model and its corresponding interpreter. AdvEdge and AdvEdge$^+$ take advantage of the edge information of the image to allow perturbation to be added to the edges in regions highlighted by the interpreters' attribution map. This enables a much stealthier attack as the generated adversarial samples are challenging to detect even with interpretation and human involvement. Moreover, the proposed attacks generate effective adversarial samples with less perturbation size.

Our Contribution. Firstly, we indicate that the existing IDLSes can be manipulated by adversarial inputs. We present two attack approaches that generate adversarial inputs to mislead the target DNN and deceive its interpreter. We evaluate our attacks against four major types of IDLSes on a dataset and compare them with the existing attack ADV^2 [17]. We summarize our contributions as follows:

- We propose AdvEdge and AdvEdge$^+$ attacks that incorporate edge information to enhance interpretation-derived attacks. We show that even restricting the perturbation to edges in regions spotted by interpretation models, the adversarial input can be very effective. We evaluate our attacks against two common DNNs architectures accompanied by four interpretation models that represent different categories.
- Our evaluation includes measuring the effectiveness of the attacks in terms of success rate and deceiving the coupled interpreters, and comparing to the

existing attack (ADV^2). The results show that the proposed attacks are as effective as ADV^2 in terms of misclassification and outperform ADV^2 in generating adversarial inputs with highly similar interpretations to their benign cases. Moreover, this level of effectiveness is maintained with a smaller amount of noise as compared with ADV^2.

Organization. The rest of the paper is organized as follows: §2 highlights the relevant literature; §3 presents the fundamental concepts; §4 and §5 describe AdvEdge and AdvEdge$^+$ attacks and their implementation against four major interpreter types; §6 shows the evaluation of the attacks' effectiveness; and §7 offers the conclusion.

2 Related Work

In this section, we provide different categories of research work that are relevant to our work: adversarial attacks and interpretability.
Attacks. Basically, there are two main threats for machine learning models: infecting the training data to weaken the target models (poisoning attack [5]) and manipulating the input data to make the target model misbehaves (evasion attack [5]). Attacking deep neural networks (DNNs) has been more challenging due to their high complexity in model architecture. Our work explores attacks against DNNs with interpretability as a defense means.
Interpretability. Interpretation models have been used to provide the interpretability for black-box DNNs via different techniques: back-propagation, intermediate representations, input perturbation, and meta models [3]. It is believed that that interpretability provides a sense of security in the decision-making process with human involvement. Nevertheless, recent work shows that some interpretation techniques are insensitive to DNNs or data generation processes, whereas the behaviors of interpretation models can be impacted significantly by the transformation without effect on DNNs [7].

Another recent work [17] shows the possibility of attacking IDLSes. Specifically, it proposes a new attacking class to deceive DNNs and their coupled interpretation models simultaneously, presenting that the enhanced interpretability provides a limited sense of security. In this work, we show the optimized version of the attack presented in the recent work [17] to deceive target DNNs and mislead their coupled interpretation models.

3 Fundamental Concepts

In this section, we introduce concepts and key terms used in the paper. We note that this paper is mainly focused on classification tasks, such as image classification. Let $f(x) = y \in Y$ denote a classifier (*i.e.,* DNN model f) that assigns an input (x) to a class (y) from a set of predefined classes (Y).

Let $g(x; f) = m$ denote an interpreter (g) that generates an attribution map (m) that reflects the importance of features in the input sample (x) based on the output of the classifier (f), (*i.e.,* the value the i-th element in m ($m[i]$) reflects the importance of the i-th element in x ($x[i]$)).

In this regard, we note that there are two main methods to achieve interpretation of a model: ❶ **Post-hoc interpretation**: The interpretation can be achieved by regulating the complexity of DNN models or by applying methods after training. This method requires creating another model to support explanations for the current model[3]. ❷ **Intrinsic interpretation**: Intrinsic interpretability can be achieved by building self-explanatory DNN models which directly integrate interpretability into their architectures [3].

Our attacks are mainly based on the first interpretation category, where an interpreter (g) extracts information (*i.e.,* attribution map m) about how a DNN model f classifies the input x.

A benign input (x) is manipulated to generate adversarial sample ($\hat{x}$) using one of the well-known attacks (PGD [9], STADV [15]) to drive the model to misclassify the input $\hat{x}$ to a target class y_t such that $f(\hat{x}) = y_t \neq f(x)$. These manipulations, *e.g.,* adversarial perturbations, are usually constrained to a norm ball $\mathcal{B}_\varepsilon(x) = \{\|\hat{x} - x\|_\infty \lneq \varepsilon\}$ to ensure its success and evasiveness. For example, PGD, a first-order adversarial attack, applies a sequence of project gradient descent on the loss function:

$$\hat{x}^{(i+1)} = \prod_{\mathcal{B}_\varepsilon(x)} \left(\hat{x}^{(i)} - \alpha .\ sign(\nabla_{\hat{x}} \ell_{prd}(f(\hat{x}^{(i)}), y_t))\right) \tag{1}$$

Here, $\prod$ is a projection operator, $\mathcal{B}_\varepsilon$ is a norm ball restrained by a pre-fixed ε, α is a learning rate, x is the benign sample, $\hat{x}^{(i)}$ is the $\hat{x}$ at the iteration i, ℓ_{prd} is a loss function that indicates the difference between the model prediction $f(\hat{x})$ and y_t.

4 AdvEdge Attack

IDLSes provide a level of security in the decision-making process with human involvement. This has been a belief until a new class of attacks is presented [17]. In their work, Zhang *et al.* [17] proposed ADV^2 that bridges the gap by deceiving target DNNs and their coupled interpreters simultaneously. Our work presents new optimized versions of the attack, namely: AdvEdge and AdvEdge$^+$. This section gives detailed information on the proposed attacks and their usage against four types of interpretation models.

4.1 Attack Definition

The main purpose of the attack is to deceive the target DNNs f and their interpreters g. To be precise, an adversarial input $\hat{x}$ is generated by adding noise to the benign input x to satisfy the following conditions:

1. The adversarial input $\hat{x}$ is misclassified to y_t by f: $f(\hat{x}) = y_t$;
2. $\hat{x}$ prompts the coupled interpreter g to produce the target attribution map m_t: $g(\hat{x}; f) = m_t$;

3. $\hat{x}$ and the benign x samples are indistinguishable.

The attack finds a small perturbation for the benign input in a way that the results of the prediction and the interpretation are desirable. We can describe the attack by the following optimization framework:

$$\min_{\hat{x}} : \Delta(\hat{x}, x) \; s.t. \begin{cases} f(\hat{x}) = y_t \\ g(\hat{x}; f) = m_t \end{cases} \tag{2}$$

As there is high non-linearity in $f(\hat{x}) = y_t$ and $g(\hat{x}; f) = m_t$ for DNNs, Equation (2) can be rewritten as the following to be more suitable for optimization:

$$\min_{\hat{x}} : \ell_{prd}(f(\hat{x}), y_t) + \lambda. \; \ell_{int}(g(\hat{x}; f), m_t) \; s.t. \; \Delta(\hat{x}, x) \leq \varepsilon \tag{3}$$

where ℓ_{prd} is the classification loss as in Equation (1), ℓ_{int} is the interpretation loss to measure the difference between the adversarial map $g(\hat{x}; f)$ and the target map m_t. To balance the two factors (ℓ_{prd} and ℓ_{int}), the hyper-parameter λ is used. We build the Equation (3) based on the PGD adversarial framework to compare the performance with the existing attack while other frameworks can also be utilized. Other settings are defined as follows: $\ell_{prd}(f(\hat{x}), y_t) = - \log(f_{y_t}(\hat{x}))$, $\Delta(\hat{x}, x) = \|\hat{x} - x\|_\infty$, and $\ell_{int}(g(\hat{x}; f), m_t) = \|g(\hat{x}; f) - m_t\|_2^2$. Overall, the attack finds the adversarial input $\hat{x}$ using a sequence of gradient descent updates:

$$\hat{x}^{(i+1)} = \prod_{\mathcal{B}_\varepsilon(x)} \left(\hat{x}^{(i)} - N_w \; \alpha. \; sign(\nabla_{\hat{x}} \ell_{adv}(\hat{x}^{(i)})) \right) \tag{4}$$

Here, N_w is the noise function that controls the amount and the position of noise to be added with respect to the benign input's edge weights w. ℓ_{adv} represents the equation of overall loss Equation (3).

AdvEdge. Notice that we apply the N_w term in Equation (4) to optimize the location and magnitude of the added perturbation. In the first attack, AdvEdge, we further restrict the added perturbation to the edges of the image that intersect with the attribution map generated by the interpreter. This means that considering the overall loss (classifier and interpreter loss), we identify the important areas of the input and then generate noise for the edges in those areas by considering the edge weights in the image obtained using the Sobel filter.

Let $\mathcal{E} : e \rightarrow \mathbb{R}^{h \times w}$ that indicates a pixel-wise edge weights matrix for an image with height h and width w, using common edge detector (*e.g.,* Sobel filters in this settings). We apply the edge weights to the sign of the gradient update as: $\mathcal{E}(x) \otimes \; \alpha. \; sign(\nabla_{\hat{x}} \ell_{adv}(\hat{x}^{(i)}))$, where $\otimes$ denotes the Hadamard product. This increases the noise on the edges while decreases the noise in smooth regions of the image. Considering $N_w = \mathcal{E}(x)$, Equation (3) can be expressed as:

$$\hat{x}^{(i+1)} = \prod_{\mathcal{B}_\varepsilon(x)} \left(\hat{x}^{(i)} - \mathcal{E}(x) \; \alpha. \; sign(\nabla_{\hat{x}} \ell_{adv}(\hat{x}^{(i)})) \right) \tag{5}$$

AdvEdge$^+$. Similar to AdvEdge, this approach also incorporates the edge weights of the input to optimize the perturbation. In this attack, we only apply the noise to the edges rather than weighting the noise in the specified areas. This is done by binarizing the edge matrix $\mathcal{E}_\delta : e \rightarrow [0,1]^{h\times w}$, . Then, obtain the Hadamard product as: $\mathcal{E}_\delta(x) \otimes \alpha.\ sign(\nabla_{\hat{x}}\ell_{adv}(\hat{x}^{(i)}))$, where the hyperparameter δ controls the threshold to binarize the edge weights. This technique allows noise values to be complete on the edges only. To improve the effectiveness of the attack, considering the restriction on the perturbation location, the threshold δ is set to 0.1. In the following subsection, we discuss the details about the attack in Equation (4) against the representatives of four types of interpretation models, namely: *back-propagation-guided interpretation*, *representation-guided interpretation*, *model-guided interpretation*, and *perturbation-guided interpretation*.

4.2 Interpretation Models

Back-Propagation-Guided Interpretation. Back-propagation-guided interpretation models calculate the gradient of the prediction of a DNN model with reference to the given input. By doing this, the importance of each feature can be derived. Based on the definition of this class interpretation, larger values in the input features indicate higher relevance to the model prediction. In this work, as the example of this class, we consider the gradient saliency (Grad) [13]. Finding the optimal $\hat{x}$ for Grad-based IDLSes is inefficient via a sequence of gradient descent updates (as in applying Equation (4)), since DNNs with ReLU activation functions cause the computation result of the Hessian matrix to be all-zero. The issue can be solved by calculating the smoothed value of the gradient of ReLU.
Representation-Guided Interpretation. In this type of interpreters, feature maps from intermediate layers of DNN models are extracted to produce attribution maps. We consider **C**lass **A**ctivation **M**ap (CAM) [18] as the representative of this class. The importance of the input regions can be identified by projecting back the weights of the output layer on the convolutional feature maps. Similar to the work of [17], we build g by extracting and concatenating attribution maps from f up to the last convolutional layer and a fully connected layer. We attack the interpreter by searching for $\hat{x}$ using gradient descent updates as in Equation (4). The attack Equation (4) can be applied to other interpreters of this class (*e.g.,* Grad-CAM [12]).
Model-Guided Interpretation. This type of interpreters trains a masking model to directly predict the attribution map in a single forward pass by masking salient positions of any input. For this type, we consider the **R**eal-**T**ime Image **S**aliency (RTS) [1]. Directly attacking RTS has been shown to be ineffective to find the desired adversarial inputs [17]. This is because the interpreter relies on both the masking model and the encoder ($enc(.)$). To overcome the issue, we add an extra loss term $\ell_{enc}(enc(\hat{x}), enc(y_t))$ to the Equation (3) to calculate the difference of the encoder's result with the adversarial input $\hat{x}$ and the target class y_t. Then, we use the sequence of gradient descent updates as defined in Equation (4) to find the optimal adversarial input $\hat{x}$.
Perturbation-Guided Interpretation. The perturbation-guided interpreters aim to find the attribution maps by adding minimum noise to the input and examining the shift in the model's output. For this work, we consider MASK [2]

as the representative of the class. As the interpreter g is constructed as optimization procedure, we cannot directly optimize the Equation (3) with the Equation (4). For this issue, bi-level optimization framework [17] can be implemented. The loss function is reformulated as: $\ell_{adv}(x,m) \triangleq \ell_{prd}(f(x),y_t) + \lambda.\ \ell_{int}(m,m_t)$ by adding m as a new variable.

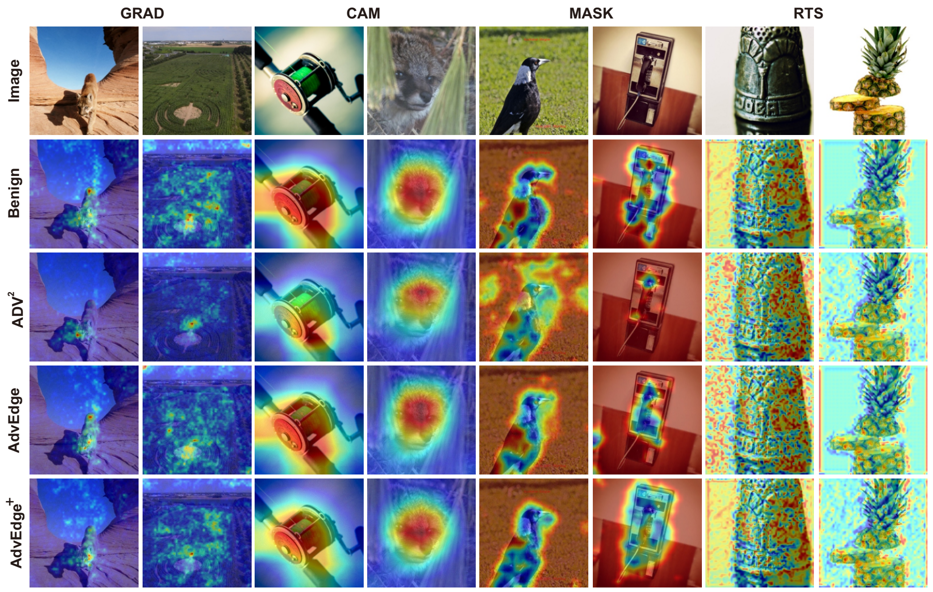


Fig. 2. Attribution maps of benign and adversarial (ADV^2, AdvEdge and AdvEdge$^+$) inputs with respect to Grad, CAM, MASK, and RTS on ResNet.

5 Experimental Setting

In this section, we explain the implementation of AdvEdge and the optimization steps to increase the effectiveness of the attacks against target interpreters (Grad, CAM, MASK, RTS). We build our two approaches (AdvEdge and AdvEdge$^+$) based on the PGD attack that utilizes the local first order information about the network (Equation (5)). For the parameters, we set $\alpha = 1./255$, and $\varepsilon = 0.031$ similar to previous studies [17]. To measure the proportion of the perturbation, ℓ_∞ is applied. To increase the efficiency of the attack, we apply a technique that adds noise to the edges of the images with a constant number of iterations (#iterations = 300). The process aims to search for perturbation points on the edges of the regions that satisfy both the classifier and interpreter.

Optimization for Both Approaches. In our attack, zero gradients of the prediction loss prevents searching the desired result with correct interpretation (Grad). To overcome the issue, the label smoothing technique with cross-entropy is proposed. In the technique, prediction loss is sampled using uniform distribution $\mathbb{U}(1-\rho,1)$ and during the attacking process, the value of ρ is decreased moderately. Considering $y_c = \frac{1-y_t}{|Y|-1}$, we calculate $\ell_{prd}(f(x),y_t) = -\sum_{c\in Y} y_c \log f_c(x)$.

Dataset. For our experiment, we use ImageNetV2 Top-Images [11] dataset. ImageNetV2 is a new test set collected based on the ImageNet benchmark and was mainly published for inference accuracy evaluation. For our test set, we use all the images that are correctly classified by the given classifier f.
Prediction Models. Two state-of-the-art DNNs are used for the experiments, **ResNet-50** [4] and **DenseNet-169** [6], which show 22.85% and 22.08% top-1 error rate on ImageNet dataset, respectively. The two DNNs are with different capacities (*i.e.,* 50 and 169 layers, respectively) and architectures (*i.e.,* residual blocks and dense blocks, respectively). Using these DNNs helps measuring the effectiveness of our attacks.
Interpretation Models. We utilize the following interpreters as the representative of Back-Propagation-Guided, Representation-Guided, Model-Guided, and Perturbation-Guided Interpretation classes: Grad [13], CAM [18], RTS [1], and MASK [2] respectively. We used the original open-source implementations of the interpreters in our experiments.
AdvEdge Attack. For the attack, we implement our attack (defined in Equation (4) on the basis on PGD framework. Other attacks frameworks (*e.g.,* STADV [15] can also be applied for the attack Equation (4). For our case, we assume that our both approaches are based on the targeted attack, in which the attack forces the DNNs to misclassify the perturbed input $\hat{x}$ to a specific and randomly-assigned target class. We compare AdvEdge and AdvEdge$^+$ with ADV^2 attack, which is considered as a new class of attacks to generate adversarial inputs for the target DNNs and their coupled interpreters. For a fair comparison, we adopt the same hyperparameters (*learning rate*, *number of iterations*, *step size*, *etc.*) and experimental settings as in ADV^2 [17].

6 Attack Evaluation

In this section, we conduct experiments to evaluate the effectiveness of AdvEdge and AdvEdge$^+$. We compare our results to ADV^2 in [17]. For this comparison, we use the original implementation of ADV^2 provided by the authors. For the evaluation, we answer the following questions: ❶ Are the proposed AdvEdge and AdvEdge$^+$ effective to attack DNNs? ❷ Are AdvEdge and AdvEdge$^+$ effective to mislead interpreters? ❸ Do the proposed attacks strengthen the attacks against interpretable deep learning models?
Evaluation Metrics. We apply different evaluation metrics to measure the effectiveness of attacks against the baseline classifiers and the interpreters. Firstly, we evaluate the attack based on deceiving the target DNNs using the following metrics:

- **Misclassification confidence:** In this metric, we observe the confidence of predicting the targeted class, which is the probability assigned by the corresponding DNN to the class y_t.

Secondly, we evaluate the attacks based on deceiving the interpreter. This is done by evaluating the attribution maps of adversarial samples. We note that this task is challenging due to the lack of standard metrics to assess the attribution maps generated by the interpreters. Therefore, we apply the following metrics to evaluate the interpretability:

- $\mathcal{L}_p$ **Measure**: We use the $\mathcal{L}_1$ distance between benign and adversarial maps to observe the difference. To obtain the results, all values are normalized to $[0, 1]$.
- **IoU Test** (**I**ntersection-**o**ver-**U**nion): This is another quantitative measure to find the similarity of attribution maps. This measurement is widely used to compare the prediction with ground truth.

Finally, to measure the amount of noise added to generate the adversarial input, the following metric is used:

- **Structural Similarity (SSIM)**: Added noise is measured by computing the mean structural similarity index [14] between benign and adversarial inputs. SSIM is a method to predict the image quality based on its distortion-free image as reference.
  To obtain the non-similarity rate (*i.e.,* distance or noise rate), we subtract the SSIM value from 1 (*i.e.,* noise_rate = 1 − SSIM).

6.1 Attack Effectiveness against DNNs

We first assess the effectiveness of AdvEdge and AdvEdge$^+$ as well as compare the results to the existing method (ADV^2 [17]) in terms of deceiving the target DNNs. We achieved 100% attack success rate of ADV^2, AdvEdge, and AdvEdge$^+$ against different classifiers and interpreters on 10,000 images.

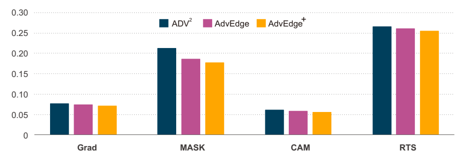
Table 1. Misclassification confidence of ADV^2, AdvEdge and AdvEdge$^+$ against different classifiers and interpreters testing on 10,000 images.

	ResNet				**DenseNet**			
	Grad	**CAM**	**MASK**	**RTS**	**Grad**	**CAM**	**MASK**	**RTS**
ADV^2	92.19%	**56.52%**	53.54%	**69.83%**	87.88%	52.88%	58.05%	57.24%
AdvEdge	**93.51%**	55.53%	**59.70%**	68.99%	**88.04%**	**53.79%**	**63.01%**	**57.26%**
AdvEdge$^+$	**92.49%**	55.40%	53.48%	69.55%	86.94%	**53.54%**	**62.82%**	**57.28%**

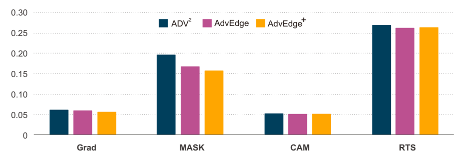
Additionally, Table 1 presents the misclassification confidence results of the three methods against different classifiers and interpreters on 10,000 images. It should be said that due to the differences in the dataset and models, the results of ADV^2 are not consistent with the results achieved in [17]. Even though our main idea is to add a small amount of perturbation to the specific regions of images, the performance is slightly better than ADV^2 in terms of Grad (ResNet), CAM (DenseNet) and RTS (DenseNet). In other cases, the results of the models' confidence are comparable.

6.2 Attack Effectiveness against Interpreters

This part evaluates the effectiveness of AdvEdge and AdvEdge$^+$ to generate similar interpretations to the benign inputs. We compare the interpretations of adversarial and benign inputs. We start with a qualitative comparison to



(a) ResNet



(b) DenseNet

Fig. 3. Average $\mathcal{L}_1$ distance of attribution maps generated by ADV^2, AdvEdge and AdvEdge$^+$ from those of corresponding benign samples on ResNet and DenseNet.

check whether attribution maps generated by AdvEdge and AdvEdge$^+$ are indistinguishable from benign inputs. By observing all the cases, AdvEdge and AdvEdge$^+$ produced interpretations that are perceptually indistinguishable from their corresponding benign inputs. As for comparing with ADV^2, all methods generated attribution maps similar to the benign inputs. Figure 2 shows a set of sample inputs together with their attribution maps in terms of Grad, CAM, RTS, and MASK. As displayed in the figure, the results of our approaches provided high similarity with their benign interpretation.

In addition to qualitative comparison, we use $\mathcal{L}_p$ to measure the similarity of produced attribution maps quantitatively. Figures 3(a) and 3(b) summarize the results of $\mathcal{L}_1$ measurement. As shown in the figures, our attacks generate adversarial samples with attribution maps closer to those generated for the benign samples compared to ADV^2. The results are similar across different interpreters on both target DNNs. We note that the effectiveness of our attack (against interpreters) varies depending on the interpreters. Generally, the results of Table 1, Figures 3(a) and 3(b) show the effectiveness of our attack in generating adversarial inputs with highly similar interpretations to their corresponding benign samples.

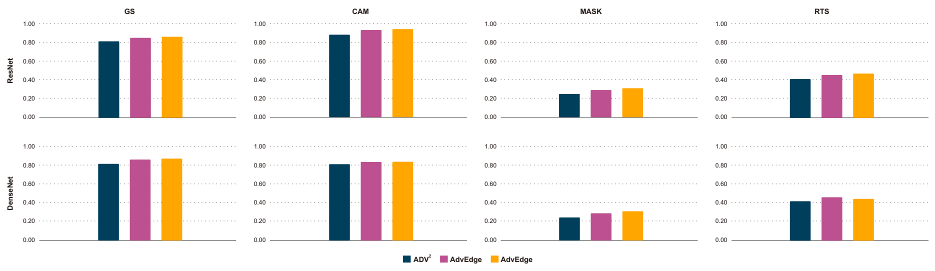
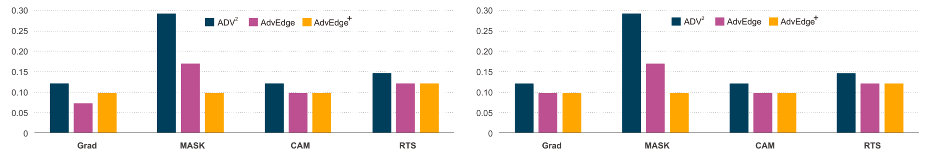


Fig. 4. IoU scores of attribution maps generated by ADV^2, AdvEdge and AdvEdge$^+$ using the four interpreters on ResNet and DenseNet. Our attacks achieve higher IoU scores in comparison with ADV^2.

Another quantitative measure to compare the similarity of attribution maps is the IoU score. As the attribution map values are floating numbers, we binarized the attribution maps to calculate the IoU. Figure 4 displays the IoU scores of attribution maps generated by ADV^2, AdvEdge and AdvEdge$^+$ using four interpreter models on ResNet and DenseNet. As shown in the figure, AdvEdge and AdvEdge$^+$ performed better than ADV^2. We note that AdvEdge$^+$ achieved



(a) ResNet (b) DenseNet

Fig. 5. Noise rate of adversarial inputs generated by ADV^2, AdvEdge and AdvEdge$^+$ on ResNet and DenseNet.

significantly better results than other methods, while AdvEdge achieved a higher score on DenseNet with RTS interpreter.

6.3 Adversarial Perturbation Rate

To measure the amount of noise added to the adversarial image by the attacks, we utilize SSIM to measure the pixel-level perturbation. Using SSIM, we infer the parts of images that are not similar to the original images, known as noise. Figures 5(a) and 5(b) display the results of comparing the noise amount generated by ADV^2, AdvEdge, and AdvEdge$^+$. The figures show that the amount of noise added using AdvEdge and AdvEdge$^+$ is significantly lower than the amount added by ADV^2. This difference is more noticeable when using the MASK interpreter. AdvEdge$^+$ adds the least amount of noise to deceive the target DNN and the MASK interpreter.

7 Conclusion

This work presents two approaches (AdvEdge and AdvEdge$^+$) to enhance the adversarial attacks on interpretable deep learning systems (IDLSes). These approaches exploit the edge information to optimize the ADV^2 attack that generates adversarial inputs to mislead the target DNNs and their corresponding interpreter models simultaneously. We demonstrated the validity and effectiveness of AdvEdge and AdvEdge$^+$ through empirical evaluation using a large dataset on two different DNNs architectures (*i.e.,* ResNet and DenseNet). We show our results against four representatives of different types of interpretation models (*i.e.,* Grad, CAM, MASK, and RTS). The results show that AdvEdge and AdvEdge$^+$ effectively generate adversarial samples that can deceive the deep neural networks and the interpretation models.
Future Work. Besides utilizing the PGD framework, this work motivates exploring other attack frameworks, such as DeepFool, STADV [8]. Another future direction is to evaluate the effectiveness of potential countermeasures to defend against AdvEdge, such as refining the DNN and interpretation models (*e.g.,* via defensive distillation [10]) or applying an adversarial sample detector (*e.g.,* feature squeezing [16] or through utilizing the interpretation transferability property in an ensemble of interpreters [17]). We also consider to test whether our approach is applicable and tractable on various sample space (*e.g.,* numerical, text, etc.).

References

1. Dabkowski, P., Gal, Y.: Real time image saliency for black box classifiers. arXiv preprint arXiv:1705.07857 (2017)
2. Fong, R.C., Vedaldi, A.: Interpretable explanations of black boxes by meaningful perturbation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3429–3437 (2017)
3. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM Comput. Surv. **51**(5) (Aug 2018)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
5. He, Y., Meng, G., Chen, K., Hu, X., He, J.: Towards security threats of deep learning systems: A survey. IEEE Transactions on Software Engineering pp. 1–1 (2020)
6. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
7. Kindermans, P.J., Hooker, S., Adebayo, J., Alber, M., Schütt, K.T., Dähne, S., Erhan, D., Kim, B.: The (un) reliability of saliency methods. In: Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, pp. 267–280. Springer (2019)
8. Laidlaw, C., Feizi, S.: Functional adversarial attacks. In: NeurIPS (2019)
9. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
10. Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks. In: 2016 IEEE symposium on security and privacy (SP). pp. 582–597. IEEE (2016)
11. Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do imagenet classifiers generalize to imagenet? In: International Conference on Machine Learning. pp. 5389–5400. PMLR (2019)
12. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. pp. 618–626 (2017)
13. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps (2014)
14. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
15. Xiao, C., Zhu, J.Y., Li, B., He, W., Liu, M., Song, D.: Spatially transformed adversarial examples. arXiv preprint arXiv:1801.02612 (2018)
16. Xu, W., Evans, D., Qi, Y.: Feature squeezing: Detecting adversarial examples in deep neural networks. arXiv preprint arXiv:1704.01155 (2017)
17. Zhang, X., Wang, N., Shen, H., Ji, S., Luo, X., Wang, T.: Interpretable deep learning under fire. In: 29th {USENIX} Security Symposium ({USENIX} Security 20) (2020)
18. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2921–2929 (2016)