Interpretable Deep Learning under Fire

Xinyang Zhang* Ningfei Wang* Shouling Ji† Hua Shen* Ting Wang⋆
*Lehigh University, {xizc15, niw217, hus218}@lehigh.edu
†Zhejiang University, sji@lehigh.edu
⋆Lehigh University, inbox.ting@gmail.com

***Abstract*—Providing explanations for complicated deep neural network (DNN) models is critical for their usability in security-sensitive domains. A proliferation of interpretation methods have been proposed to help end users understand the inner workings of DNNs, that is, how a DNN arrives at a particular decision for a specific input. This improved interpretability is believed to offer a sense of security by involving human in the decision-making process. However, due to its data-driven nature, the interpretability itself is potentially susceptible to malicious manipulation, about which little is known thus far.**

In this paper, we conduct the first systematic study on the security of interpretable deep learning systems (IDLSes). We first demonstrate that existing IDLSes are highly vulnerable to adversarial manipulation. We present ACID attacks, a broad class of attacks that generate adversarial inputs which not only mislead target DNNs but also deceive their coupled interpretation models. By empirically investigating three representative types of interpretation models, we show that ACID attacks are effective against all of them. This vulnerability thus seems pervasive in many IDLSes. Further, using both analytical and empirical evidence, we identify the prediction-interpretation "independency" as one possible root cause of this vulnerability: a DNN and its interpretation model are often not fully aligned, resulting in the possibility for the adversary to exploit both models simultaneously. Moreover, by examining the transferability of adversarial inputs across different interpretation models, we expose the fundamental tradeoff among the attack evasiveness with respect to different interpretation methods. These findings shed light on developing potential countermeasures and designing more robust interpretation methods, leading to several promising research directions.

I. Introduction

The recent advances in machine learning (ML), especially deep learning [29], have led to breakthroughs in a number of long-standing artificial intelligence tasks (e.g., image classification [23], [57], [63], natural language processing [62], [67], and even playing Go [55]), enabling use cases previously considered strictly experimental.

However, the state-of-the-art performance of deep neural network (DNN) models is often achieved at the expense of interpretability: it is often challenging to intuitively and quantitatively understand the inference of complicated DNNs– how does a DNN arrive at a particular decision for a specific input – due to their high non-linearity and nested structures. This is a major drawback for applications where the interpretability of decisions is a critical prerequisite. For instance, incorrect predictions can be consequential for medical diagnosis [37], autonomous driving [8], and financial services [51]; therefore, simple black-box predictions cannot be trusted by default. Another major drawback of DNNs is their inherent vulnerability

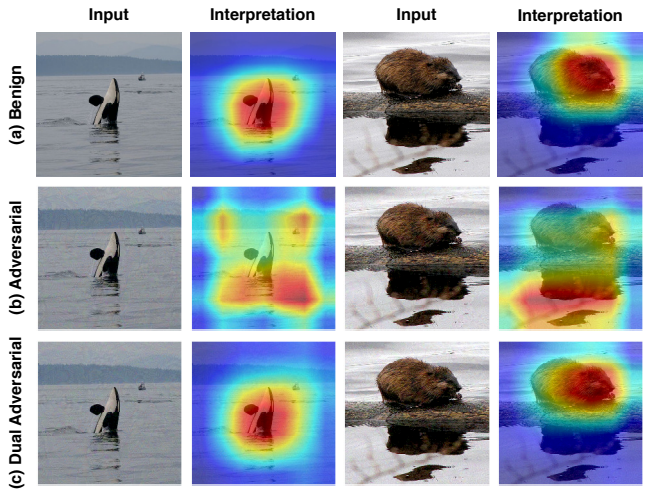

Figure 1: (a) benign, (b) regular adversarial, and (c) ACID adversarial inputs and interpretation, with RESNET [23] and CAM [72] as the classifier and interpreter respectively.

to adversarial inputs – those maliciously crafted samples to trigger target DNNs to misbehave [64], [10], [28] – which often leads to unpredictable model behavior and hinders their application in security-sensitive domains [8], [51], [27].

The above drawbacks have spurred intensive research on improving the interpretability of DNNs and ML models in general, through providing explanations at either model-level [26], [50], [5], [71] or instance-level [56], [18], [48], [12]. For instance, in Fig. 1 (a), the attribution map highlights the most informative part of an image with respect to its classification, revealing the causal relationship between the input and the model prediction. This interpretability helps ML developers and operators better understand the inner workings of DNNs, enabling use cases such as model debugging [44], digesting security analysis results [21], and detecting adversarial inputs [15]. For instance, in Fig. 1 (b), an adversarial input [36], which causes a target DNN to derail from its normal behavior, often generates interpretation drastically different from its benign counterpart, and is thereby easily detectable.

As illustrated in Fig. 2, a DNN model (classifier), coupled with an interpretation model (interpreter), forms an interpretable deep learning system (IDLS). Compared with regular deep learning systems, the enhanced interpretability of IDLSes is believed to offer a sense of security by involving human in the decision-making process [65]. However, given its data-driven nature, the interpretability itself can potentially become

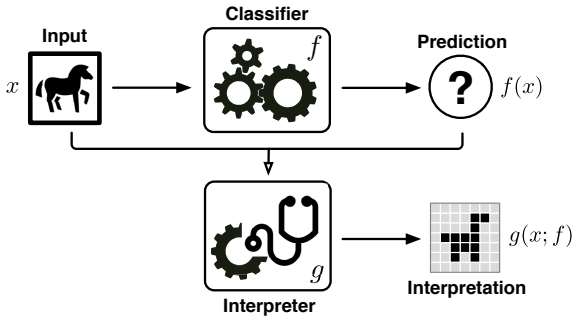

Figure 2: Flow of interpretable deep learning system (IDLS).

the target of malicious manipulation. Unfortunately, thus far, little is known about the security vulnerability of IDLSes, not to mention mitigating such threats.

Our Work: To bridge this striking gap, in this paper, we conduct an in-depth study on the security of IDLSes, which leads to the following interesting findings.

Foremost, we demonstrate that existing IDLSes are highly vulnerable to adversarial attacks. We present ACID attacks[1], a broad class of attacks that generate adversarial inputs to mislead not only the target classifier but also its coupled interpreter. By empirically evaluating ACID attacks on three major classes of interpretation methods, we show that generating adversarial inputs deceiving both the classifier and its interpreter is not significantly more difficult than producing adversarial inputs fooling the classifier only. For instance, Fig. 1 (c) shows adversarial inputs which are misclassified by RESNET [23] and also interpreted highly similarly to their benign counterparts. Thus, the improved interpretability of IDLSes merely provides limited security assurance.

Further, we conduct in-depth analysis on the fundamental causes of this vulnerability. Using both empirical and analytical evidence, we show that one possible reason may be the partial "independency" between prediction and interpretation: the interpreter's interpretation only partially describes the classifier's prediction, allowing the adversary to exploit both models simultaneously. This finding entails several intriguing and important questions: (i) What is the root cause of this independency? (ii) What is its implication for the vulnerability of different interpretation methods? (iii) What is its implication for designing more robust interpretation methods? We explore all these questions in our study.

Finally, we investigate the transferability of ACID adversarial inputs across different interpreters, which complements the study on the transferability of adversarial attacks across DNNs [45], [33]. It is observed that it is difficult to find adversarial inputs transferable across different classes of interpreters, as they construct interpretation from distinct perspectives (e.g., intermediate representations [53], input-prediction correspondences [18], and meta models [12]). This finding points to training an ensemble of multiple, complementary interpreters as one promising countermeasure against ACID attacks.

[1] ACID: Adversarial Classification and Interpretation Duality.

Notation	Definition
f	target classifier
g	target interpreter
x_0	benign input
x_*	adversarial input
t	adversary's target class
δ	perturbation vector
$x[i]$	i-th dimension of x
ϵ	perturbation magnitude threshold
$\|\cdot\|$	vector norm

Table I. Symbols and notations.

Our Contributions: To the best of our knowledge, this work represents the first systematic study on the security of IDLSes. Our contributions can be summarized as follows.

- By presenting and evaluating effective attacks against three major classes of interpreters, we demonstrate that the interpretability of existing IDLSes only provides limited security assurance.
- With extensive empirical and analytical analysis, we identify the prediction-interpretation independency as one possible cause for this vulnerability, which raises concerns regarding today's assessment metrics of interpretation methods. Further, by comparing the vulnerability of different interpreters, we reveal the intricate conflict between visual interpretability and attack robustness.
- Through exploring the transferability of adversarial inputs across different classes of interpreters, we point to training an ensemble of multiple, complementary interpreters as a promising direction of improving the robustness of IDLSes against adversarial manipulation.

We believe these findings will shed light on designing and operating IDLSes in a more secure and informative manner.

Roadmap: The remainder of the paper proceeds as follows. §II introduces a set of fundamental concepts; §III presents ACID attacks and details the concrete implementation against three major classes of interpretation methods; §IV empirically evaluates ACID attacks; §V discusses the root cause of vulnerability, explores the transferability of ACID adversarial inputs, and points to future research directions; §VI surveys relevant literature; The paper is concluded in §VII.

II. PRELIMINARIES

We begin with introducing a set of fundamental concepts and assumptions. The symbols and notations used in the paper are summarized in Tab. I.

Classifiers – In this paper, we primarily focus on predictive tasks (e.g., image classification [14], sentiment analysis [59]), in which a DNN model f classifies a specific input x into a set of predefined classes $\mathcal{C}$, $f(x) = c \in \mathcal{C}$.

Interpreters – In general, the interpretability can be obtained in two ways: designing interpretable DNNs [70], [50] or extracting post-hoc interpretation for existing DNNs. As it does not modify the model architecture, the latter case typically leads to higher prediction accuracy. Thus, we mainly

consider post-hoc interpretation in the following. Further, we focus on instance-level interpretation [26], [50], [42], [12], [54], [71], [56], [18], [41], which explains how f classifies a specific input x and uncovers the causal relationship between the input x and the prediction $f(x)$. Specifically, we assume such explanations are given in the form of *attribution maps*. As shown in Fig. 2, the interpreter g generates an explanation $m = g(x; f)$, with its i-th dimension $m[i]$ quantifying the importance of x's i-th feature $x[i]$ for the prediction $f(x)$.

Adversarial Attacks – DNNs are inherently vulnerable to adversarial inputs, which are maliciously crafted to force target DNNs to misbehave [13], [64], [39]. Specifically, an adversarial input x_* is often generated by slightly modifying a benign input x_0, with the objective of forcing f to misclassify x_* into a target class t, $f(x_*) = t \neq f(x_0)$. Here we focus on the setting of targeted attacks. To ensure the attack evasiveness, the perturbation is constrained to a set of allowed perturbations (e.g., a norm ball $\mathcal{B}_\epsilon(x_0) = \{x | \|x - x_0\|_\infty \leq \epsilon\}$).

Without loss of generality, in this paper, we consider the PGD attack [36], a universal first-order adversarial attack, as the reference attack model. At a high level, the PGD attack is implemented as a sequence of project gradient descent steps on the negative loss function:

$$x^{(i+1)} = \Pi_{x_0+\mathcal{B}_\epsilon}\left(x^{(i)} - \alpha \operatorname{sign}\left(\nabla_x \ell_{\text{prd}}\left(f(x), t\right)\right)\right) \quad (1)$$

where Π denotes the projection operator, α ($\alpha \geq 0$) represents the learning rate, the loss function ℓ_{prd} measures the difference of the model prediction $f(x)$ and the class t desired by the adversary (e.g., cross entropy), and $x^{(0)}$ is initialized as x_0.

Interpretation as Defenses – The interpreter g allows the user to intuitively examine f's behavior. An adversarial input x_* tends to cause f to derail from its normal behavior, resulting in interpretation $m_* = g(x_*; f)$ significantly deviating from its benign counterpart $m_0 = g(x_0; f)$, $m_0 \not\approx m_*$ (see Fig. 1(b)). Thus, the user may use interpretation as a means to detect erroneous or adversarial inputs [65], [21].

III. ACID ATTACK

While it is believed that the interpretability of IDLSes offers a sense of security by involving the user into the decision-making process [65], [21], [19], [15], this belief has not been rigorously tested. In this paper, we challenge this conventional wisdom by presenting the ACID attack, a broad class of attacks that deceive target DNN models and their interpreters simultaneously.

In the following, we first present an overview of ACID attack, elaborate on its instantiation against each of three major classes of interpreters, and then discuss the implementation details.

A. Attack Overview

In the ACID attack, the adversary attempts to create adversarial inputs that deceive both the DNN model f and its interpreter g. To this end, she creates the adversarial input x_* by perturbing a benign input x_0 such that (i) $f(x_*) = t$, which ensures that x_* is misclassified as desired by the adversary; and (ii) $g(x_*; f) \approx g(x_0; f)$, which indicates that x_0 and x_* share similar interpretation from the user perspective. In other words, the goal is to find sufficiently small perturbation to the benign input that changes its classification but keeping its interpretation intact.

At a high level, we formulate the ACID attack using the following optimization framework:

$$\begin{aligned} \min_{\delta} \quad & \ell_{\text{int}}\left(g(x; f), m_0\right) \\ \text{s.t.} \quad & \begin{cases} f(x) = t \\ m_0 = g(x_0; f) \\ \|x - x_0\|_\infty \leq \epsilon \end{cases} \end{aligned} \quad (2)$$

where the interpretation loss ℓ_{int} quantifies the difference of the benign and adversarial attribution maps $m_0 = g(x_0; f)$ and $m_* = g(x_*; f)$, which can be instantiated using ℓ_p norm. The constraints ensure that (i) the adversarial input is misclassified as desired by the adversary and (ii) the perturbation is constrained within the norm ball $\mathcal{B}_\epsilon(x_0)$.

While the formulation in Eq. (2) seems intuitive, solving it directly for concrete interpreters entails non-trivial challenges. The interpreter g often employs complicated mechanisms. For instance, in MASK [18], g itself is formulated using an optimization framework. Further, the constraint $f(x_*) = t$ is highly non-linear for practically useful DNNs.

We thus reformulate Eq. (2) as an appropriate instance that can be solved using existing optimization algorithms:

$$\begin{aligned} \min_{\delta} \quad & \ell_{\text{prd}}(f(x), t) + \lambda \ell_{\text{int}}\left(g(x; f), m_0\right) \\ \text{s.t.} \quad & \begin{cases} m_0 = g(x_0; f) \\ \|x - x_0\|_\infty \leq \epsilon \end{cases} \end{aligned} \quad (3)$$

where ℓ_{prd} is the same as that defined in Eq. (1) and the parameter λ balances the two factors. We thus define the overall adversarial loss as: $\ell_{\text{adv}}(x) = \ell_{\text{prd}}(f(x), t) + \lambda \ell_{\text{int}}\left(g(x; f), m_0\right)$, where we omit the benign input x_0 and target class t for simplicity.

Given the use of ℓ_∞ norm to constrain the perturbation magnitude, we construct the optimization solution on top of the PGD attack framework [36]. In the following, we detail the instantiation of ACID attack against each of three major classes of interpreters.

B. Representation-guided Interpretation

The first class of interpreters leverage the feature maps at intermediate layers of DNNs to generate the attribution maps. We use Class Activation Mapping (CAM) [72] as a concrete example to illustrate the ACID attack against this class of interpreters.

At a high level, CAM performs global average pooling [30] over the feature maps of the last convolutional layer and uses the outputs as features for a linear layer with softmax activation to approximate the model prediction. Based on this connectivity structure, CAM computes the attribution maps

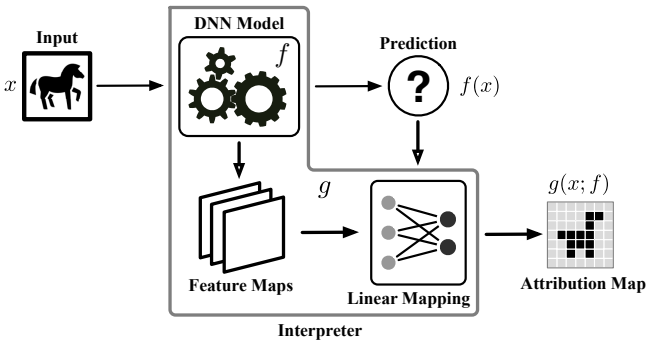

Figure 3: Flow of representation-guided interpretation.

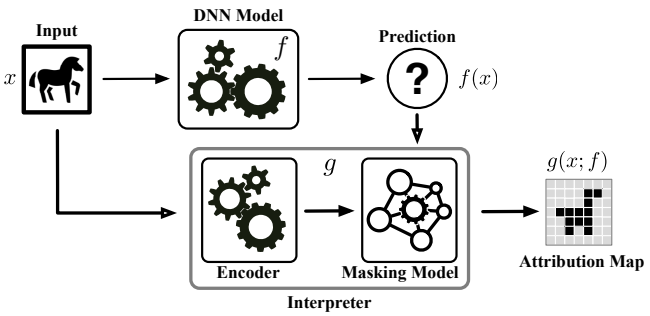

Figure 4: Flow of model-guided interpretation.

by projecting the weights of the linear layer back onto the convolutional feature maps, which is illustrated in Fig. 3.

Formally, let $a_k[i,j]$ represent the activation of the k-th channel of the last convolutional layer at the spatial position (i,j). The output of global average pooling is defined as $A_k = \sum_{i,j} a_k[i,j]$. Thus, the probability $f_c(x)$ for the class c with respect to the input x is approximated by:

$$f_c(x) \approx \sum_k w_{k,c}\, A_k = \sum_{i,j} \sum_k w_{k,c}\, a_k[i,j] \tag{4}$$

Thus, the class activation map m_c is given by:

$$m_c[i,j] = \sum_k w_{k,c}\, a_k[i,j] \tag{5}$$

Due to its use of deep representations at intermediate layers, CAM is often able to generate attribution maps of high visual quality and limited noise and artifacts.

Thus, we instantiate g with a DNN that concatenates the part of f until the last convolutional layer and a linear layer with its parameters configured by $\{w_{k,c}\}$, and define the interpretation loss as $\ell_{\text{int}}(g(x;f), m_0) = \|g(x;f) - m_0\|_2^2$. We then find x_* using a sequence of gradient descent updates:

$$x^{(i+1)} = \Pi_{\mathcal{B}_\epsilon(x_0)} \left(x^{(i)} - \alpha \operatorname{sign}\left(\nabla_x \ell_{\text{adv}}(x)\right) \right) \tag{6}$$

We also extend this attack to the case of GRADCAM [53], another representation-guided interpreter. The details are deferred to the appendix.

C. Model-guided Interpretation

Instead of relying on the deep representations at intermediate layers, model-guided methods train a meta-model to directly predict an attribution map for any given input in a single feed-forward pass, which is illustrated in Fig. 4. We consider RTS [12] as a representative method in this category.

For an given input x in the class c, RTS finds its attribution map m by solving the following optimization problem:

$$\begin{aligned} \min_m \quad & \lambda_1 r_{\text{tv}}(m) + \lambda_2 r_{\text{av}}(m) - \log\left(f_c\left(\phi(x;m)\right)\right) \\ & + \lambda_3 f_c\left(\phi(x;1-m)\right)^{\lambda_4} \\ \text{s.t.} \quad & 0 \le m \le 1 \end{aligned} \tag{7}$$

Here $r_{\text{tv}}(m)$ represents the total variation of m, which reduces the noise and artifacts in m; $r_{\text{av}}(m)$ represent the average value of m, which minimizes the size of retained parts; $\phi(x;m)$ is the operator of using m as a mask to blend x with random colors and Gaussian blur, which measure the impact of retained parts (where $m = 1$) on the model prediction; and the parameters λ_i $(i = 1, \cdots, 4)$ balances these factors. Intuitively, the formulation above is designed to find the sufficient and necessary parts of x, based on which f is able to make the prediction $f(x)$ with high confidence.

However, solving Eq. (7) for each input during test time is fairly expensive. Instead, one may train a DNN to directly predict the attribution map m for each input x, without accessing to the DNN f after training. In [49], this is achieved by composing a ResNet [23] pre-trained on ImageNet [14] as the encoder (which extracts feature maps at different scales) and a U-NET [49] as the masking model, which is then trained to optimize Eq. (7).

We thus consider the composition of the pre-trained encoder and the trained masking model as the interpreter g and define the interpretation loss as: $\ell_{\text{int}}(g(x;f), m_0) = \|g(x;f) - m_0\|_2^2$. However, our evaluation shows that directly minimizing ℓ_{int} is often ineffective for finding the desired adversarial input. This may be explained by that the encoder enc plays a significant role in generating the attribution map, while solely relying on the output of the masking model is insufficient to guide the attack. We thus add to Eq. (3) an additional loss term $\ell_{\text{prd}}(\text{enc}(x), \text{enc}(x_0))$, which quantifies the difference of the predictions of benign and adversarial inputs by the encoder.

We find the adversarial input x_* using a sequence of gradient descent updates similar to Eq. (6). The implementation details are deferred to §III-E.

D. Perturbation-guided Interpretation

The third class of interpreters formulate finding the attribution map by perturbing the input with minimum noise and observing the change of the model prediction. We use MASK [18] as a representative model in this class to illustrate the implementation of ACID attack.

For a specific input x, MASK identifies its most informative parts by checking whether changing such parts influences the prediction $f(x)$, as illustrated in Fig. 5. It formulates an optimization framework that learns a perturbation mask m, where $m[i] = 0$ if the i-th input feature is retained and $m[i] = 1$ if the feature is replaced with Gaussian noise. The

optimal mask is found by solving the optimization problem:

$$\begin{aligned} \min_{m} \quad & f_c\left(\phi(x;m)\right) + \lambda \|1 - m\|_1 \\ \text{s.t.} \quad & 0 \le m \le 1 \end{aligned} \tag{8}$$

where c is the current prediction $c = f(x)$ and $\phi(x;m)$ is the perturbation operator which blends x with Gaussian noise. The first term finds m that causes the probability of the current prediction to decrease significantly, while the second term encourages m to be sparse. Intuitively, solving Eq. (18) amounts to finding highly informative and necessary parts of x with respect to its prediction $f(x)$. Note that the formulation above may result in significant artifacts in m. A refined formulation is deferred to the appendix.

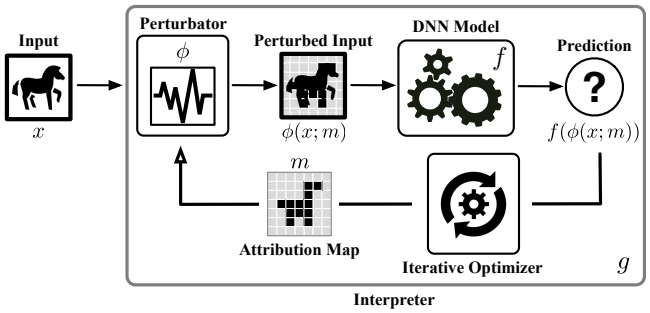

Figure 5: Flow of perturbation-guided interpretation.

Let $m_0 = g(x_0; f)$ denote the attribution map of the benign input x_0 with respect to its current prediction $f(x)$, and $m_* = g(x; f)$ be the map of the adversarial input x with respect to the target class t. We again define the interpretation loss in Eq. (3) as $\ell_{\text{int}}(x) = \|g(x;f) - m_0\|_2^2$. However, unlike the cases of representation- and model-guided interpretation, in the current case, it is infeasible to directly optimize Eq. (3) using gradient descent Eq. (6), as the interpreter g itself is formulated as an optimization problem.

Instead, we reformulate the ACID attack using a bilevel optimization framework. For given x_0, t, f, and g, we redefine the adversarial loss function in Eq. (3) as $\ell_{\text{adv}}(x, m) = \ell_{\text{prd}}(f(x), t) + \lambda \ell_{\text{int}}(m, m_0)$ by introducing m as an additional variable. We further define $\ell_{\text{map}}(m; x)$ as the objective function defined in Eq. (18). Note that $m_*(x) = \arg\min_m \ell_{\text{map}}(m, x)$ is the attribution map found by MASK for the adversarial input x. We then have the following attack framework:

$$\begin{aligned} \min_{x} \quad & \ell_{\text{adv}}\left(m_*(x), x\right) \\ \text{s.t.} \quad & m_*(x) = \arg\min_{m} \ell_{\text{map}}(m, x) \end{aligned} \tag{9}$$

Still, solving the bilevel optimization in Eq. (9) exactly is expensive, as it requires recomputing $m_*(x)$ by solving the inner optimization problem whenever x is updated. We propose an approximate iterative procedure which optimizes x and m by alternating between gradient descent on ℓ_{adv} and ℓ_{map} respectively.

More specifically, at the i-th iteration, given the current input $x^{(i-1)}$, we compute the attribution map $m^{(i)}$ by updating $m^{(i-1)}$ with gradient descent on $\ell_{\text{map}}\left(m^{(i-1)}, x^{(i-1)}\right)$; we then fix $m^{(i)}$ and obtain $x^{(i)}$ by minimizing ℓ_{adv} after a single step of gradient descent with respect to $m^{(i)}$. Formally, we define the objective function for updating $x^{(i)}$ as:

$$\ell_{\text{adv}}\left(m^{(i)} - \xi \nabla_m \ell_{\text{map}}\left(m^{(i)}, x^{(i-1)}\right), x^{(i-1)}\right)$$

where ξ is the learning rate for the virtual gradient descent.

The rationale behind this design is that while it is difficult to directly minimizing $\ell_{\text{adv}}\left(m_*(x), x\right)$ with respect to x, we use a single-step unrolled map to serve as the surrogate of $m_*(x)$. A similar approach has been used in [17]. Essentially, this iterative optimization defines a Stackelberg game between the optimizer for x (leader) and the optimizer for m [52] (follower), which requires the leader to anticipate the follower's next move to reach the equilibrium.

Algorithm 1: ACID attack against MASK.

Input: x_0: benign input; t: target class; f: target DNN; g: MASK interpreter
Output: x_*: adversarial input

initialize x and m as x_0 and $g(x_0; f)$;
while *not converged* **do**
  `// update` m
  update m by gradient descent along $\nabla_m \ell_{\text{map}}(m, x)$;
  `// update` x `with single-step lookahead`
  update x by gradient descent along $\nabla_x \ell_{\text{adv}}\left(m - \xi \nabla_m \ell_{\text{map}}\left(m, x\right), x\right)$;
return x;

Algorithm 1 sketches the ACID attack against MASK. More implementation details are deferred in §III-E. The theoretical justification for Algorithm 1 is given in the appendix.

E. Implementation and Optimization

Next we detail the concrete implementation of ACID attack and present a suite of optimization strategies to improve the attack effectiveness against specific interpreters.

Iterative Optimizer – As we use ℓ_∞ norm to constrain the perturbation magnitude, we construct the iterative optimizer on top of the PGD attack framework [36], which updates the adversarial input at the i-th step as:

$$x^{(i+1)} = \Pi_{\mathcal{B}_\epsilon(x_0)}\left(x^{(i)} - \alpha \,\text{sign}\left(\nabla_x \ell_{\text{adv}}(x)\right)\right) \tag{10}$$

Note that it is also possible to construct the optimizer using alternative frameworks (e.g., the C&W attack [10]) if other metrics (e.g., ℓ_2 norm) are considered.

Warm Start – It is observed in our empirical evaluation that it is often inefficient to search for adversarial inputs by running the update steps of ACID attack (i.e., Eq. (10)) from the scratch. Instead, first running a fixed number (e.g., 400) of update steps of PGD attack and then resuming the ACID update steps significantly improves the search efficiency. Intuitively, this strategy first quickly approaches the manifold of adversarial inputs, then searches for inputs that satisfy both prediction and interpretation constraints.

Multistep Lookahead – In implementing Algorithm 1, we apply multiple gradient descent steps in both updating m (line 3) and computing the surrogate map $m_*(x)$ (line 4), which is observed to lead to faster convergence in our empirical evaluation. Further, to improve the optimization stability, we may use the average gradient to update m. Specifically, let $\{m_j^{(i)}\}$ be the sequence of maps obtained at the i-th iteration by applying multistep gradient descent. We then use the aggregated interpretation loss $\sum_j \|m_j^{(i)} - m_0\|_2^2$ to compute the average gradient for updating m.

Adaptive Learning Rate – Besides using multistep gradient descent, to improve the convergence efficiency, we also dynamically adapt the learning rate for updating both m and x. To update m, at each iteration, we use a running Adam optimizer as a meta-learner [3] to estimate the optimal learning rate for updating m (line 3). We update x in a two-step fashion to stabilize the training: (i) first updating x in terms of adversarial loss, and (ii) updating it in terms of interpretation loss. During (ii), we use a binary search to find the largest step size, such that x's confidence is still above a certain threshold c_{min} after perturbation.

Periodical Reset – Recall that in Algorithm 1, we update the estimate of attribution map by following gradient descent on ℓ_{map}. As the number of update steps increases, this estimate may deviate significantly from the true map generated by the interpreter, which negatively impacts the attack effectiveness. To address this, periodically (e.g., every 50 iterations), we replace the estimated map with the map $g(x; f)$ that is directly computed by the interpreter based on the current adversarial input. At the same time, we reset Adam step to correct its internal state.

IV. Evaluation

In this section, we conduct an empirical evaluation of the ACID attack on a variety of DNN and interpreters from both qualitative and quantitative perspectives. Specifically, our experiments are designed to answer the following critical questions.

- **Q:** How effective is the ACID attack in terms of deceiving target DNN models?
  A: We show that the ACID attack achieves attack success rate comparable with regular adversarial attacks across different DNNs, indicating that despite its more complicated objective, the ACID attack is as effective as regular adversarial attacks.
- **Q:** How effective is the ACID attack in terms of generating highly plausible interpretation?
  A: We show that the attribution maps of dual adversarial inputs (generated by the ACID attack) and benign inputs are highly indistinguishable, under various measures including ℓ_p distance, intersection-over-union test, and human user study.
- **Q:** How evasive is the ACID attack with respect to existing adversarial attack detection mechanisms?
  A: We show that the ACID attack is no more detectable than regular adversarial attacks with respect to state-of-the-art detection methods, implying their fundamentally similar attack evasiveness.

Next we first introduce the setting of our experiments.

A. Experimental Setting

Datasets – We run all the experiments on the IMAGENET [14] (ILSVRC 2012 competition) dataset, which consists of 1.2 million hand labeled images in 1,000 classes. With 10-crop, every image is of 224×224 pixels. In our experiments, with respect to each classifier, we randomly sample 1,000 images from the validation set of the IMAGENET dataset that are classified correctly by the classifier initially.

Classifiers – We consider two state-of-the-art DNNs as the classifiers, ResNet [23] (ResNet-50) and DenseNet [25] (DenseNet-169), which respectively attain 22.85% and 22.08% top-1 error on IMAGENET. Using two DNNs of different complexities (50 layers versus 169 layers) and architectures (residual blocks versus dense blocks), we intend to factor out the influence of the characteristics of individual models.

Interpreters – We consider the three interpreters presented in §III (CAM [72], RTS [12], and MASK [18]) as the representatives of representation-, model-, and perturbation-guided interpreters respectively. We adopt their open-source implementation in our experiments. As the original RTS implementation is tightly coupled with the target DNN (i.e., ResNet), we train a new encoder for the DenseNet model,

	Parameter Setting
CAM	regularizer λ = 25.0
MASK	gradient descent steps per iteration n_{step} = 5
	iterations per reset n_{reset} = 50
	maximum binary search step size $\alpha_{\max}$ = 0.08
	maximum binary search step n_{bs} = 15
	Adam hyper-parameters $\alpha = 0.1$, $\beta_1 = 0.9$, $\beta_2 = 0.999$
RTS	regularizer λ = 0.23 (ResNet) 1.07 (DenseNet)

Table II. Setting of hyper-parameters for the ACID attack.

Attacks – We implement all the instances of ACID attack in §III within a projected gradient descent framework. For all the instances, we have the following default parameter setting: total number of iterations n_{total} = 1,200, number of warmup iterations $n_{\text{start}} = 400$, and learning rate α = 0.01. Tab. II lists the instance- and DNN-specific parameter setting. For comparison, we also implement the regular PGD attack [36], a universal first-order attack, for which we use the following parameter setting: total number of iterations n_{total} = 1,100 and learning rate α = 1/255. By default, for both the ACID and PGD attacks, we assume the setting of targeted attacks, in which the adversary desires to force the target DNN to misclassify the given adversarial input into a randomly designated class.

B. Attack Effectiveness (Prediction)

In the first set of experiments, we evaluate the effectiveness of ACID attack in terms of deceiving target DNN models.

The attack effectiveness is measured using *attack success rate*, which is defined as:

$$\text{Attack Success Rate} = \frac{\#\text{successful trials}}{\#\text{total trials}}$$

and *misclassification confidence*, which is the probability assigned by the DNN to the class desired by the adversary.

	ResNet			DenseNet		
	CAM	MASK	RTS	CAM	MASK	RTS
PGD	98.00% (0.99)			97.67% (0.99)		
ACID	96.33% (0.99)	96.33% (0.98)	97.67% (0.98)	97.67% (0.99)	95.00% (0.98)	96.00% (0.98)

Table III. Attack effectiveness of ACID and PGD against various combinations of classifiers and interpreters.

Tab. III summarizes the attack success rates and misclassification confidence of ACID and PGD against various combinations of classifiers and interpreters. Note that as PGD is only applied on the classifier, its effectiveness is agnostic to different interpreters. Here, to make fair comparison, we fix the maximum number of iterations = 1,000 for both attacks. It is observed that ACID achieves high success rate (above 95%) and misclassification confidence (above 0.98) across all the settings, which is comparable with the regular PGD attack. We thus have the following conclusion:

Observation 1

Despite its more complicated objective, the ACID attack is as effective as regular adversarial attacks in terms of deceiving target DNNs.

C. Attack Effectiveness (Interpretation)

In this set of experiments, we evaluate the effectiveness of ACID in terms of generating highly plausible interpretation. In specific, we compare the similarity of interpretation of adversarial (by ACID) and benign cases. Due to the lack of standard definition for interpretation similarity, we use a variety of measures to comprehensively compare the interpretation of adversarial and benign inputs.

Visualization – We first qualitatively compare the interpretation of benign and adversarial (generated by PGD and ACID) inputs. Fig. 6 visualizes a set of sample images and their attribution maps generated by CAM, MASK, and RTS (more samples are given in the appendix). Observe that across all the interpreters, the ACID adversarial inputs generate interpretation perceptually indistinguishable from their benign counterparts. In comparison, the PGD adversarial inputs can be easily identified by inspecting their attribution maps.

ℓ_p Distance – Besides qualitatively comparing the attribution maps of benign and adversarial inputs, we also measure their similarity quantitatively. By considering attribution maps as matrices, we measure the ℓ_p distance between benign and adversarial (by PGD and ACID) attribution maps using their ℓ_p distance. Tab. IV summarizes the results. Note that here we normalize all the measures to [0, 1] for comparison, for the attribution maps of CAM, MASK, and RTS are of size 7×7, 28×28, and 56×56 respectively.

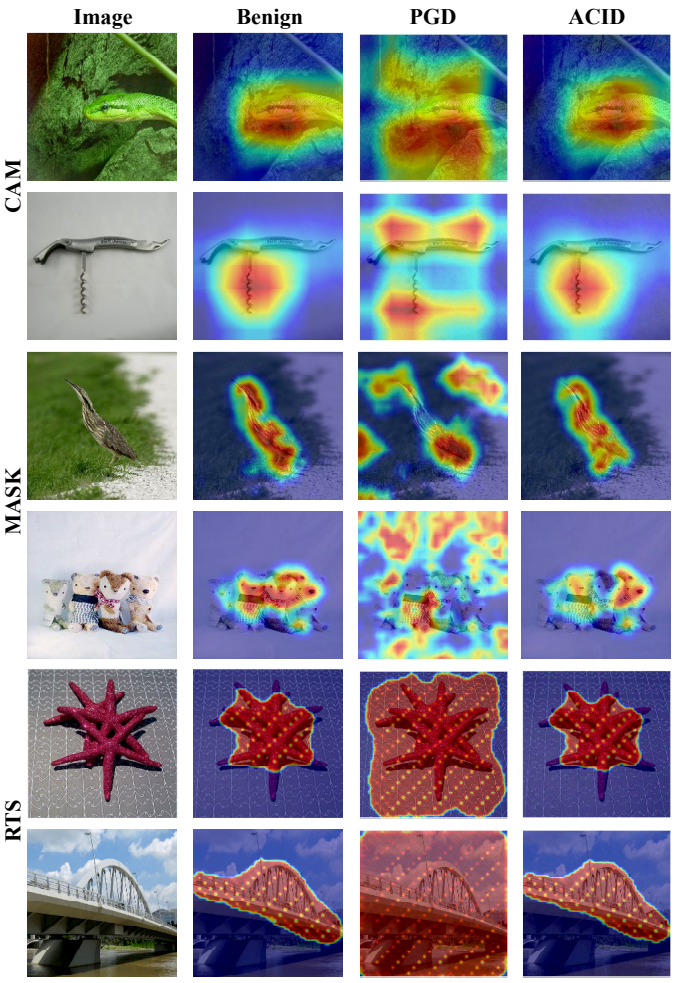

Figure 6: Visualization of attribution maps of benign and adversarial (by the PGD and ACID attacks) inputs with respect to CAM, MASK, and RTS on ResNet.

	ResNet		DenseNet	
	ℓ_p $(p=1)$	ℓ_p $(p=2)$	ℓ_p $(p=1)$	ℓ_p $(p=2)$
CAM-P	0.26 ± 0.06	0.32 ± 0.07	0.26 ± 0.11	0.32 ± 0.12
CAM-A	0.08 ± 0.07	0.10 ± 0.09	0.06 ± 0.02	0.08 ± 0.02
MASK-P	0.22 ± 0.11	0.33 ± 0.10	0.22 ± 0.11	0.32 ± 0.11
MASK-A	0.09 ± 0.04	0.19 ± 0.05	0.08 ± 0.05	0.18 ± 0.05
RTS-P	0.68 ± 0.14	0.79 ± 0.10	0.26 ± 0.14	0.48 ± 0.15
RTS-A	0.02 ± 0.01	0.09 ± 0.04	0.03 ± 0.02	0.11 ± 0.09

Table IV. ℓ_p distance of attribution maps of benign inputs and adversarial inputs by PGD (-P) and ACID (-A) attacks.

We have the following observations. (i) ACID is able to generate adversarial inputs with attribution maps highly similar to benign cases. In comparison, the attribution maps of benign and adversarial inputs by PGD are significantly different. For instance, the average ℓ_1 distance of ACID is more than 60% smaller than that of PGD across all three interpreters on ResNet. (ii) The effectiveness of ACID varies across different interpreters. For instance, it achieves the lowest ℓ_1 distance

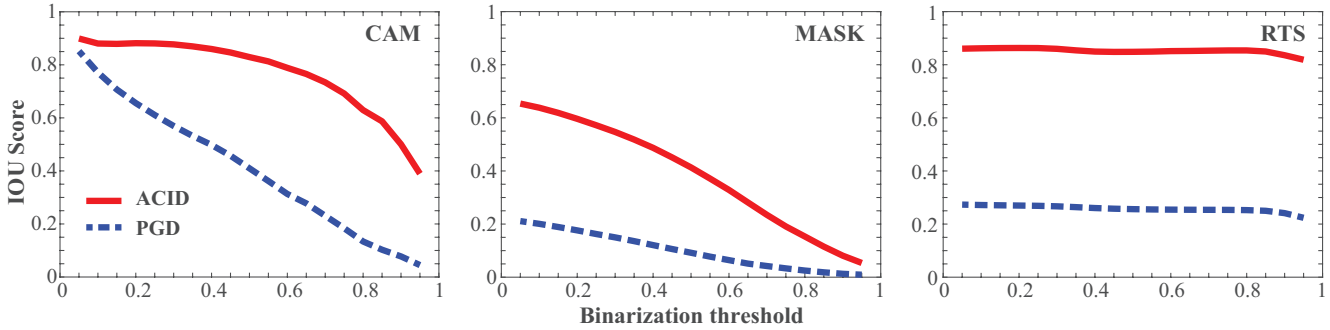

Figure 7: IOU scores of adversarial attribution maps (by PGD and ACID) with respect to benign maps on ResNet.

on RTS, which is about 78% smaller than that on MASK, indicating that different interpreters feature varying robustness against the attack. (iii) The effectiveness of ACID seems insensitive to the DNN model. On both ResNet and DenseNet, ACID attains fairly similar ℓ_p distance measures.

IOU Test – Another quantitative measure for the similarity of attribution maps is the intersection-over-union (IOU) score. It is widely used in object detection [22] to compare model predictions with ground-truth bounding boxes. Formally, the IOU score of a binary-valued map m with respect to a baseline map m_0 is defined as their Jaccard similarity:

$$\mathrm{IOU}(m) = \frac{|\mathsf{O}(m) \cap \mathsf{O}(m_0)|}{|\mathsf{O}(m) \cup \mathsf{O}(m_0)|} \tag{11}$$

where $\mathsf{O}(m)$ denotes the set of non-zero dimensions in m. In our case, as the values of attribution maps are floating numbers, we use a threshold to binarize them.

Fig. 7 shows the average IOU scores of adversarial inputs (by PGD and ACID) with respect to benign inputs under varying binarization threshold. Observe that with proper threshold setting, ACID attains much higher IOU scores than PGD (i.e., more than 0.4 across all the cases), especially in the case of RTS, in which the attribution maps are natively binary-valued.

User Study – The ultimate measure of interpretability is given by human users. We conduct a user study on the similarity of benign and adversarial (by ACID) attribution maps. We design the following game. The user is presented with a randomly sampled pair (x, m), where x is a benign image and m is its attribution map (either benign or adversarial), and required to label m as either benign (negative) or adversarial (positive). We measure the perceptual difference of benign and adversarial maps by the accuracy that the users successfully distinguish benign and adversarial cases.

	ResNet		DenseNet	
	Precision	Recall	Precision	Recall
CAM	0.59 ± 0.07	0.55 ± 0.08	0.59 ± 0.06	0.60 ± 0.09
MASK	0.62 ± 0.06	0.63 ± 0.08	0.66 ± 0.08	0.64 ± 0.08
RTS	0.54 ± 0.07	0.52 ± 0.09	0.59 ± 0.09	0.57 ± 0.07

Table V. Separability of benign and adversarial (by ACID) attribution maps for human users.

We recruit 26 voluntary students on campus, and each is presented with 10 randomly sampled image-map pairs (5 benign and 5 adversarial cases) on each interpreter. Tab. V lists the average precision and recall attained by the participants. Observe that the scores are close to random guess, implying that benign and adversarial maps are highly indistinguishable. Further, among the three interpreter, the adversarial cases on MASK are the most identifiable, while that on RTS are the least discernible. We defer the discussion on the inherent connection between interpretability and vulnerability to §V.

Based on the experiments above, we can have the following conclusion.

Observation 2

It is practical to generate adversarial inputs with highly plausible interpretation from both qualitative and quantitative perspectives.

D. Attack Evasiveness

Next we evaluate the evasiveness of ACID with respect to state-of-the-art adversarial attack defense mechanisms.

Regular ACID – To be succinct, in our study, we use Feature Squeezing (FS) [69], a state-of-the-art adversarial defense method. Intuitively, FS reduces the adversary's search space by coalescing many inputs corresponding to different feature vectors into a single input, and detects adversarial inputs by comparing their predictions under original and squeezed settings. The coalescing operations are implemented using a set of "squeezers": bit depth reduction – reducing the bit length of each pixel, local smoothing – replacing each pixel with the median value of neighboring pixels, and non-local smoothing – replacing each image patch with the average of similar patches. Further, one may use an ensemble of multiple squeezers to improve the detection effectiveness.

Tab. VI lists the detection rates of adversarial inputs (by PGD and ACID) using different types of squeezers as well as the optimal ensemble squeezer on ResNet. It is observed that the detection rates of ACID and PGD inputs by the optimal ensemble squeezer differs by less than 5%.

Observation 3

The overall detectability of adversarial inputs generated by ACID and regular attacks is not much different.

Squeezer	Setting	PGD	CAM-A	MASK-A	RTS-A	CAM-A*	MASK-A*	RTS-A*
Bit Depth Reduction	1-bit	83.7%	78.4%	59.9%	74.6%	40.3%	7.6%	47.7%
	2-bit	86.7%	87.5%	84.1%	92.1%	37.7%	11.7%	40.7%
	3-bit	43.2%	56.1%	89.2%	73.6%	8.8%	35.9%	14.0%
	4-bit	0.7%	1.0%	51.3%	2.1%	0.0%	28.8%	0.4%
	5-bit	0.3%	0.0%	19.0%	0.0%	0.0%	18.0%	0.0%
Local Smoothing	2×2	50.7%	70.7%	96.4%	88.7%	34.7%	81.7%	40.7%
	3×3	78.9%	89.2%	98.6%	95.8%	14.7%	16.5%	14.0%
Non-Local Smoothing	11-3-2	2.7%	5.2%	38.6%	9.9%	7.0%	30.6%	3.5%
	11-3-4	22.1%	34.8%	74.7%	55.7%	21.2%	51.7%	17.9%
	13-3-2	3.1%	6.6%	39.7%	10.7%	7.0%	31.7%	3.5%
	13-3-4	25.7%	39.0%	77.9%	59.8%	23.1%	54.1%	22.1%
Ensemble (Setting)		89.5% (2,2×2,11-3-4)	90.2% (2,2×2,11-3-4)	98.9% (3,3×3,11-3-4)	94.5% (2,2×2,11-3-4)	43.2% (2,2×2,13-3-4)	81.2% (4,2×2,11-3-2)	44.9% (2,2×2,11-3-4)

Table VI. Detectability of adversarial inputs by PGD, regular ACID (-A), and adaptive ACID (-A*) using feature squeezing.

Yet, the effectiveness of individual squeezers vary across different cases. For instance, the ACID inputs against MASK seem more discernible than that against CAM and RTS from the perspective of local smoothing. This difference may be explained by the inherent connection between interpretability and vulnerability (details in §V).

Algorithm 2: Adaptive ACID against bit depth reduction.

Input: x_0: benign input; t: target class; f: target DNN; g: MASK interpreter; ψ: bit depth reduction
Output: x_*: adversarial input

```
// attack in squeezed space
```
$x' \leftarrow$ PGD attack on $\psi(x_0)$ with target t;
```
// attack in original space
```
search for $x_* = \arg\min_{x \in \psi^-(x')} \ell_{\text{adv}}(x)$;
return x_*;

Adaptive ACID – We then raise this question: is it possible for ACID to evade the detection of FS? To this end, we use two optimization strategies to enhance the evasiveness of ACID. Related to existing adaptive attacks against FS [24], this optimization is interesting in its own right. Specifically, for local and non-local smoothing squeezers, we introduce into the optimization objective of ACID (Eq. (3)) another adversarial loss term $\ell_{\text{sqz}} = \|f(x) - f(\psi(x))\|$, which measures the difference of predictions for the original input x and squeezed input $\psi(x)$ (ψ is the squeezer). For bit depth reduction, our attack consists of three main steps. (i) We first run the regular untargeted PGD attack over the squeezed input $\psi(x_0)$ and find the adversarial input x'. Note that in the squeezed space, the perturbation δ is also squeezed, i.e., $x' = \psi(x_0) + \psi(\delta)$. (ii) We then project x' back to the original space by inverting the squeezing operation ψ^-. Because bit depth reduction is a many-to-one operation, $\psi^-(x')$ represents a set of inputs in the original space. (iii) We search in $\psi^-(x')$ for the adversarial input x_* that minimizes the attack objective of ACID. The overall algorithm is sketched in Algorithm 2.

Tab. VI shows the evasiveness of adversarial inputs generated by adaptive ACID (A*). Compared with regular ACID inputs, the detection rates decrease by 47.0%, 17.7%, and 49.6% for adaptive ACID inputs on CAM, MASK, and RTS respectively. Note that here we only show the possibility of improving the evasiveness of ACID attacks against representative defense methods. We consider an in-depth study of this matter as our ongoing work. Meanwhile, we measure the ℓ_1 distance between the attribution maps of benign and adaptive ACID inputs, which are listed in Tab. VII. The comparison with Tab. IV shows that the optimization in adaptive ACID has little impact on its attack effectiveness against interpreters. We may thus conclude:

	ℓ_p $(p=1)$	ℓ_p $(p=2)$
CAM	0.09 ± 0.03	0.11 ± 0.04
MASK	0.09 ± 0.04	0.15 ± 0.06
RTS	0.04 ± 0.03	0.14 ± 0.07

Table VII. ℓ_p distance of benign and adaptive ACID maps.

Observation 4

It is possible to generate adversarial inputs that are both plausible in terms of interpretability and evasive in terms of detectability.

V. DISCUSSION

While it is shown that ACID is effective against a range of interpreters, the cause of this vulnerability is unclear yet. Next, we conduct an in-depth study on this root cause from both analytical and empirical perspectives, and further explore potential countermeasures based on our findings. Specifically, our study answers the following key questions.

- **Q:** What are possible causes of this vulnerability?
  A: We identify the "independency" between prediction and interpretation as one potential cause: the interpreter is only partially aligned with the classifier (i.e., DNN), allowing the adversary to exploit both models simultaneously.
- **Q:** What are possible causes of this independency?
  A: We show that the over-reliance on visual assessment may partially account for this phenomenon: the visual quality of interpretation outweighs its fidelity of describing DNN behavior in assessing an interpreter.

- **Q:** What are possible countermeasures against ACID?
  A: We show that as different interpreters focus on distinct aspects of DNN behaviors, the ensemble of multiple, complementary interpreters tends to make launching the ACID attack prohibitively difficult.

A. Root of Attack Vulnerability

Recall that the formulation of ACID in Eq. (3) defines two seemingly conflicting objectives, maximizing the change of prediction while minimizing the change of interpretation. We thus conjecture that the ACID attack is feasible because the classifier and its interpreter are partially independent of each other: the interpreter's interpretation only partially describes the classifier's prediction, making it possible for the adversary to exploit both models simultaneously. For instance, MASK focuses on the input-prediction correspondences while ignoring the internal representations; CAM relies on the deep representations at intermediate layers, while neglecting the input-prediction correspondences; RTS uses both the internal representations in an auxiliary encoder and the input-interpretation correspondences in training data, which however may deviate from the true behavior of DNN models.

Targeted Interpretation – To validate this proposition, we consider a variant of ACID attacks with targeted interpretation. For a given input x_0, we randomly generate a target class t and a target interpretation m_t, and then search for an adversarial input x_* that triggers the DNN to misclassify it as t and also generates interpretation similar to m_t (i.e., $f(x_*) = t$ and $g(x_*; f) \approx m_t$). Intuitively, if ACID is able to find such x_*, it indicates that the DNN and its interpreter can be manipulated separately; in other words, they are only partially aligned with each other.

CAM	MASK	RTS
96.00%	98.67%	96.00%

Table VIII. Effectiveness of ACID with targeted interpretation.

For a given input, we generate the target attribution map by (i) sampling a patch of random shape (either a rectangle or a circle), random angle, and random position over the input, and (ii) setting the elements inside the patch as '1' and those outside it as '0'. A set of sample inputs and target maps are illustrated in Fig. 8. It is observed that the target maps tend to significantly deviate from the benign maps. To implement this variant of ACID attack, we simply replace the benign map m_0 in Eq. (3) with the target map m_t.

Tab. VIII lists the attack success rates of ACID with targeted interpretation. Note that compared with Tab. III, the targetedness of interpretation has little influence on the attack effectiveness, implying that the space of adversarial inputs is sufficiently large to contain ones with targeted interpretation.

Fig. 8 visualizes a set of sampled adversarial inputs produced by ACID with targeted interpretation on CAM, MASK, and RTS. It is observed that across all the cases the adversarial maps appear visually similar to the target maps, highlighting the attack effectiveness in terms of generating targeted interpretation. This is further quantitatively validated in Tab. IX. Observe that the ℓ_p distance between ACID and target maps is comparable with that between ACID and benign maps in Tab. IV. Based on the experiments above, we have the following observation.

	w.r.t. target map		w.r.t. benign map	
	ℓ_p $(p=1)$	ℓ_p $(p=2)$	ℓ_p $(p=1)$	ℓ_p $(p=2)$
CAM	0.08 ± 0.02	0.12 ± 0.03	0.50 ± 0.05	0.58 ± 0.04
MASK	0.15 ± 0.05	0.25 ± 0.07	0.42 ± 0.07	0.52 ± 0.05
RTS	0.09 ± 0.04	0.18 ± 0.08	0.49 ± 0.07	0.65 ± 0.05

Table IX. ℓ_p distance between ACID and target maps and that between ACID and benign maps.

Observation 5

A DNN and its interpreter are often partially independent of each other, allowing the adversary to manipulate both models simultaneously.

It is noted that using an analogy with edge detection in images, prior work [1] empirically shows that some existing interpretation methods seem insensitive to both the DNN models and the data generation process. In this paper, we cast new light on the independency between prediction and interpretation by exploring this phenomenon from the perspective of adversarial vulnerability.

B. Root of Prediction-Interpretation Independency

Next we explore the causes for this prediction-interpretation independency. We conjecture that one possible reason lies in that the current assessment of interpretation methods overly relies on *visual interpretability*, which prefers interpretation that visually agrees with human expectation [1], while downplaying its fidelity of describing DNN behavior.

Intuitively, as it triggers abnormal DNN behavior, an adversarial input tends to cause significant interpretation change, if the interpretation faithfully reflects DNN behavior. Thus, the connection between visual interpretability and interpretation fidelity is translated into the tradeoff between visual interpretability and attack robustness. This tradeoff is observed in our empirical evaluation (§IV). For instance, MASK, compared with CAM and RTS, generates interpretation with much more noise and artifacts, and is thereby considered inferior in visual assessment [2]; yet, as it better captures the DNN behavior by directly modeling the input-prediction correspondences, MASK is less vulnerable to the ACID attack than CAM and RTS, which can be observed in Tab. IV, Tab. V, and Tab. VI.

We further demonstrate the conflict between visual interpretability and attack robustness by exploring a family of interpretation methods that are natively robust against the ACID attack and yet are often considered inferior due to their low visual interpretability.

Back-Propagation Interpretation – Back-propagation based interpreters calculate the gradient (or its variant) of a neuron with respect to a specific input and derive the importance of each feature [56], [58], [54], [2]. Intuitively, the gradient magnitude of a feature indicates its relevance

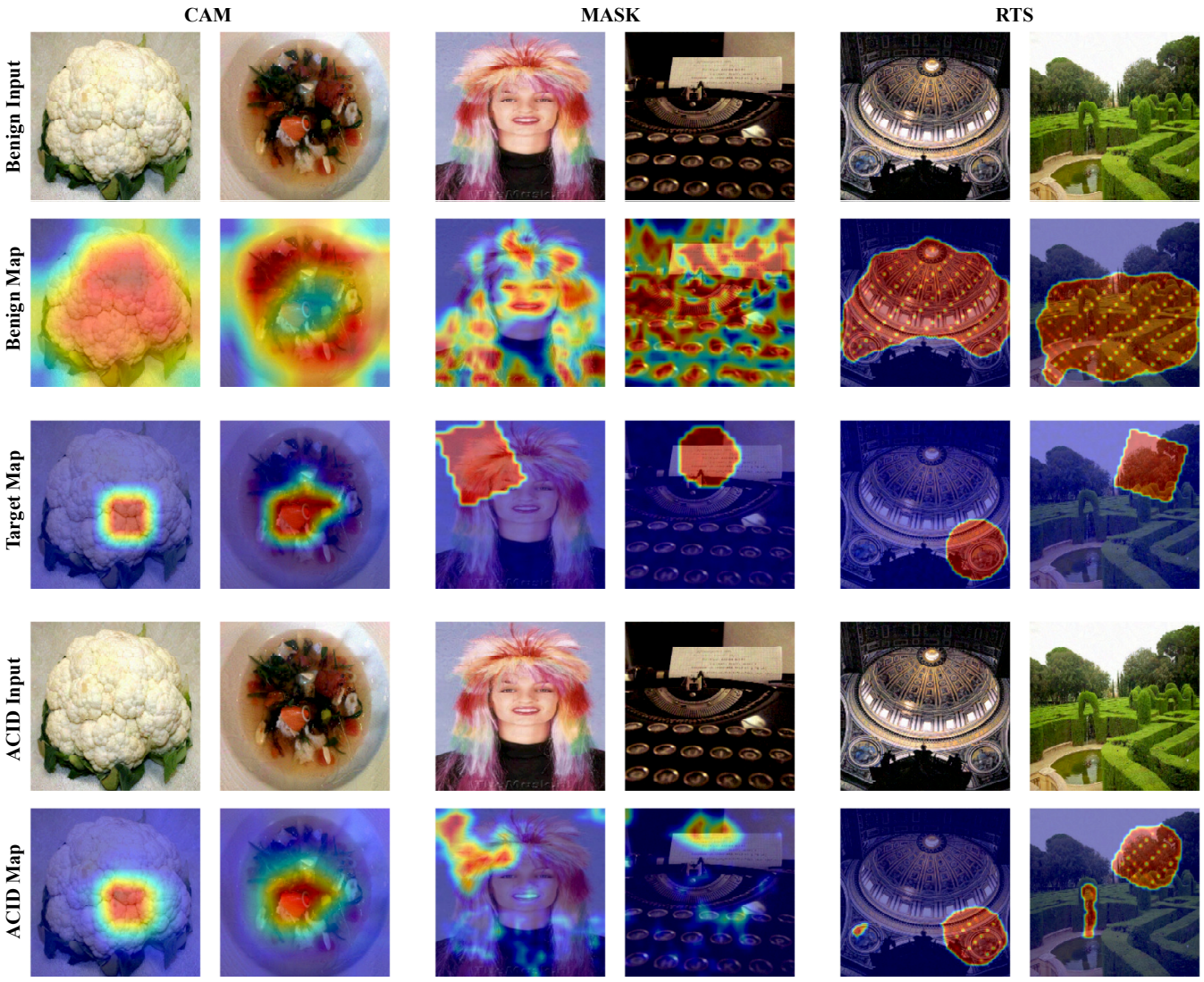

Figure 8: Visualization of ACID attacks targeting randomly generated attribution maps with respect to CAM, MASK, and RTS.

to the prediction. Limited by their heuristic nature, back-propagation-based methods often generate interpretation of low visual quality (e.g., noise and artifacts) and may highlight irrelevant features. Thus, they are often considered inferior to other methods with better visual assessment results [53], [72], [71], [12].

However, unlike other methods which are shown highly vulnerable to the ACID attack (§IV), back-propagation-based methods are fundamentally robust against such attacks. Next we formally prove this notion. Without loss of generality, we use integrated gradient [61] (IG) as a representative back-propagation method in our discussion.

At a high level, for the i-th feature of a given input x, IG computes its attribution $m[i]$ by aggregating the gradient of $f(x)$ along the path from a baseline input x_0 to x:

$$m[i] = (x[i] - x_0[i]) \int_0^1 \frac{\partial f(xt + (1-t)x_0)}{\partial x[i]} \mathrm{d}t \qquad (12)$$

Note that IG satisfies the desirable *completeness* axiom [54] that the attributions sum up to the difference between the prediction of f at the input x and the baseline x_0. All other back-propagation interpretation methods share similar formulations [2].

Attack Robustness – To simplify the exposition, let us assume a binary classification setting with classes {+, -}. The DNN models a function f which predicts the probability of x from the positive class as $f(x)$, and $1 - f(x)$ for the negative class respectively. Given a specific input x, which belongs to the negative class, the adversary creates an adversarial input $x_* = x + \delta$ and attempts to trigger f to misclassify x_* as positive. We model the adversarial prediction loss as $\ell_{\text{prd}}(\delta) = f(x_*) - f(x)$, i.e., the increase in the probability of positive prediction. Using integration, we compute the

prediction loss as follows:

$$\ell_{\text{prd}}(\delta) = \int_0^1 \nabla f(tx_* + (1-t)x)^\top (x_* - x) \mathrm{d}t \qquad (13)$$

Meanwhile, we define the adversarial interpretation loss as $\ell_{\text{int}}(\delta) = \|m - m_*\|$, where m and m_* are the attribution maps of x and x_* respectively. While it is difficult to directly quantify $\ell_{\text{int}}(\delta)$, we use the attribution map of x_* with x as the baseline as a surrogate:

$$\Delta m[i] = (x_*[i] - x[i]) \int_0^1 \frac{\partial f(x_* t + (1-t)x)}{\partial x_*[i]} \mathrm{d}t \qquad (14)$$

which quantifies the impact of the i-th feature on the difference between the predictions $f(x)$ and $f(x_*)$. We thus have $\ell_{\text{int}}(\delta) = \|\Delta m\|_1$.

Proposition 1. *The prediction loss is upper bounded by the interpretation loss as:* $\ell_{\text{prd}}(\delta) \leq \ell_{\text{int}}(\delta)$.

Proof. We define u as the input difference $u = (x_* - x)$ and v as the integral vector with its i-th element $v[i]$ as

$$v[i] = \int_0^1 \frac{\partial f(x_* t + (1-t)x)}{\partial x_*[i]} \mathrm{d}t$$

According to the definitions, we have: $\ell_{\text{prd}}(\delta) = u^\top v$ and $\ell_{\text{int}}(\delta) = \|u \odot v\|_1$ where $\odot$ is the Hadamard product.

We have the following derivation: $\ell_{\text{prd}}(\delta) = \sum_i u[i]v[i] \leq \sum_i \|u[i] \cdot v[i]\| = \ell_{\text{int}}(\delta)$. Therefore, the prediction loss is upper-bounded by the interpretation loss. □

In other words, in order to force x_* to be misclassified with high confidence, the difference of the benign and adversarial attribution maps needs to be large. Recall that the objectives of ACID attack is to maximize the prediction loss while minimizing the interpretation loss. The coupling between prediction and interpretation losses results in a fundamental conflict for the ACID attack. Similar reasoning applies to other back-propagation based methods given their equivalence [2].

With the empirical and analytical evidence above, we have the following observation.

Observation 6

There exists an inherent conflict between visual interpretability and attack robustness.

C. Possible Countermeasures

Finally, we discuss potential countermeasures against the ACID attack. Motivated by the observation that different interpreters focus on distinct aspects of DNN behavior (e.g., CAM focuses on deep representations while MASK focuses on input-prediction correspondences), we then explore the *transferability* of adversarial inputs across different interpreters.

One intriguing property of adversarial inputs is their transferability – an adversarial input effective against one DNN is often found effective against another DNN, even though it is not crafted on the second one [45], [33], [39]. In this set of experiments, we investigate whether such transferability phenomena exist in attacks against interpreters; that is, whether an adversarial input that generates plausible interpretation against one interpreter is also able to generate probable interpretation against another interpreter.

In this experiment, for each given interpreter g, we randomly select a set of adversarial inputs crafted against g (source) and compute their interpretation on another interpreter g' (target). Fig. 9 illustrates the attribution maps of a given adversarial input on g and g'. Further, for each case, we compare the adversarial map (right) against the corresponding benign map (left). It is observed that the interpretation transferability is fairly low: an adversarial input crafted against one interpreter g rarely generates highly plausible interpretation on another interpreter g'. We further quantitatively validate these observations. Tab. X measures the ℓ_p distance between adversarial and benign maps across different interpreters. For comparison, we also show the ℓ_p distance for adversarial inputs generated by PGD. It is observed that an adversarial input crafted on g tends to generate low-quality interpretation on a different interpreter g', with quality comparable to that generated by interpretation-agnostic attacks (e.g., PGD). We can thus conclude:

Observation 7

The transferability of adversarial inputs across different complementary interpreters is low.

Based on this observation, a promising direction to defend against the ACID attack is to deploy multiple, complementary interpreters to provide a holistic view of the behavior of DNN models. We consider the exploration of ensemble defenses as our ongoing research.

VI. Related Work

In this section, we survey three categories of work related to adversarial inputs, namely, attacks and defenses, transferability, and interpretability.

Attacks and Defenses – Due to their use in security-critical domains, machine learning models are increasingly becoming the targets of malicious attacks [7]. Two primary threat models are considered in literature: Poisoning attacks – the adversary pollutes the training data to eventually compromise the target models [6], [68], [40]; Evasion attacks – the adversary modifies the input data during inference to trigger target models to misbehave [13], [34], [43].

Compared with simple machine learning models (e.g., decision tree, support vector machine, logistic regression), securing deep neural network (DNN) models deployed in adversarial settings poses even more challenges due to their significantly higher model complexity [29]. One line of work focuses on developing new evasion attacks against DNN models [64], [20], [47], [10], [36]. Another line of work attempts to improve DNN resilience against such attacks by inventing new training and inference strategies [46], [38], [69], [35]. Yet, such defenses are often circumvented by even powerful attacks [10]

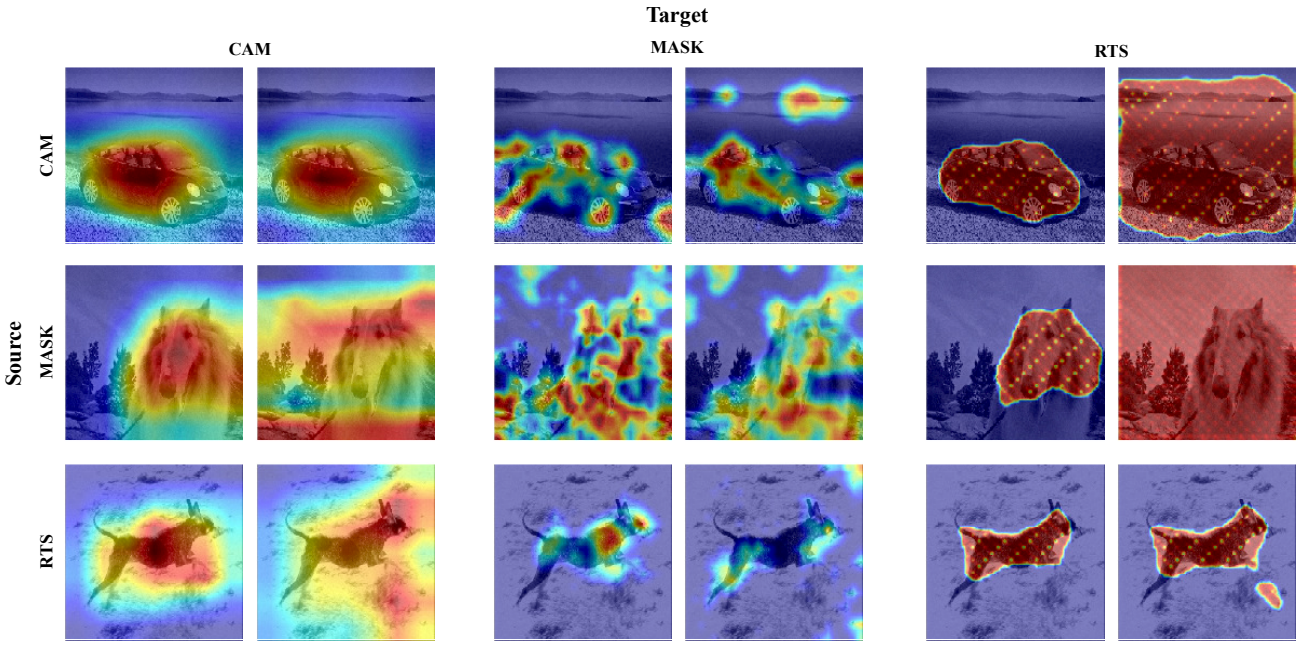

Figure 9: Visualization of attribution maps of adversarial inputs across different interpreters.

	CAM		MASK		RTS	
	ℓ_1	ℓ_2	ℓ_1	ℓ_2	ℓ_1	ℓ_2
CAM	0.08 ± 0.07	0.10 ± 0.09	0.21 ± 0.09	0.32 ± 0.08	0.55 ± 0.19	0.70 ± 0.15
MASK	0.34 ± 0.09	0.40 ± 0.09	0.09 ± 0.04	0.19 ± 0.05	0.74 ± 0.13	0.84 ± 0.09
RTS	0.20 ± 0.06	0.25 ± 0.08	0.22 ± 0.09	0.33 ± 0.09	0.02 ± 0.01	0.09 ± 0.04
PGD	0.26 ± 0.06	0.32 ± 0.07	0.22 ± 0.11	0.33 ± 0.10	0.68 ± 0.14	0.79 ± 0.10

Table X. ℓ_p distance between attribution maps of adversarial and benign inputs across different interpreters.

or adaptively engineered adversarial inputs [9], [4], resulting in a constant arms race between adversaries and defenders [31].

This work is among the first to explore attacks against DNN models with interpretability as a means of defense.

Transferability – One intriguing property of adversarial attacks is their transferability [64]: adversarial inputs crafted against one DNN model is often found effective against another model. This property enables black-box adversarial attacks: the adversary is able to generate adversarial inputs based on a surrogate model and apply them on the target model [45], [11], [33]. To defend against such attacks, the method of ensemble adversarial training [66] has been proposed recently, which trains a DNN model using data augmented with adversarial inputs crafted on other models.

This work complements this line of work by investigating the transferability of adversarial inputs across different classes of interpreters.

Interpretability – A proliferation of methods have been proposed in literature to provide interpretability for black-box machine learning models, using techniques based on back-propagation [56], [60], [61], intermediate representations [72], [53], [16], perturbation [18], and meta models [12].

The improved interpretability is believed to offer a sense of security by involving human in the decision-making process. Recent work has exploited interpretability to debug DNN models [44], digest security analysis results [21], and especially detect adversarial inputs [32], [65]. Specifically, as adversarial inputs cause unexpected behaviors of DNN models, the interpretation of DNN behaviors is expected to differ significantly between benign and adversarial inputs.

This work shows the possibility of deceiving a DNN model and its interpreter simultaneously, implying that the improved interpretability only provides limited security assurance, which complements prior work on examining the fidelity of existing interpretation methods [1] from the perspective of adversarial vulnerability. The findings point to the necessity of designing and operating interpretable deep learning in a more secure and informative manner.

VII. Conclusion

This work represents a systematic study on the security of interpretable deep learning systems (IDLSes). We present ACID attacks, a broad class of attacks to generate adversarial inputs that not only mislead target DNN models but also deceive corresponding interpretation models. Through extensive empirical evaluation, we show that ACID attacks are effective against a wide range of DNNs and interpretation models, implying that the interpretability of existing IDLSes only offers a false sense of security. We identify the prediction-

interpretation independency as one possible root cause of this vulnerability, raising the critical concern regarding the current assessment metrics of interpretation methods. Further, we show that there exists fundamental tradeoff between the attack evasiveness with respect to different interpretation mechanisms, which sheds light on developing potential countermeasures and designing more robust interpretation methods.

REFERENCES

[1] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity Checks for Saliency Maps," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2018.

[2] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross, "Towards Better Understanding of Gradient-based Attribution Methods for Deep Neural Networks," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.

[3] M. Andrychowicz, M. Denil, S. Gómez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. de Freitas, "Learning to learn by gradient descent by gradient descent," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2016.

[4] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.

[5] O. Bastani, C. Kim, and H. Bastani, "Interpretability via Model Extraction," in *Workshop on Fairness, Accountability, and Transparency in Machine Learning (FAT/ML)*, 2017.

[6] B. Biggio, B. Nelson, and P. Laskov, "Poisoning Attacks against Support Vector Machines," in *Proceedings of IEEE Conference on Machine Learning (ICML)*, 2012.

[7] B. Biggio and F. Roli, "Wild Patterns: Ten Years after the Rise of Adversarial Machine Learning," *Pattern Recognition*, vol. 84, pp. 317–331, 2018.

[8] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to End Learning for Self-Driving Cars," *ArXiv e-prints*, 2016.

[9] N. Carlini and D. Wagner, "Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods," in *Proceedings of ACM Workshop on Artificial Intelligence and Security (AISec)*, 2017.

[10] N. Carlini and D. A. Wagner, "Towards Evaluating the Robustness of Neural Networks," in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, 2017.

[11] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "Zoo: Zeroth Order Optimization based Black-Box Attacks to Deep Neural Networks without Training Substitute Models," in *Proceedings of ACM Workshop on Artificial Intelligence and Security (AISec)*, 2017.

[12] P. Dabkowski and Y. Gal, "Real Time Image Saliency for Black Box Classifiers," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2017.

[13] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma, "Adversarial Classification," in *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, 2004.

[14] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[15] M. Du, N. Liu, Q. Song, and X. Hu, "Towards Explanation of DNN-based Prediction with Guided Feature Inversion," in *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, 2018.

[16] M. Du, N. Liu, Q. Song, and X. Hu, "Towards Explanation of DNN-based Prediction with Guided Feature Inversion," *ArXiv e-prints*, 2018.

[17] C. Finn, P. Abbeel, and S. Levine, "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks," in *Proceedings of IEEE Conference on Machine Learning (ICML)*, 2017.

[18] R. C. Fong and A. Vedaldi, "Interpretable Explanations of Black Boxes by Meaningful Perturbation," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017.

[19] T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, and M. Vechev, "AI2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation," in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, 2018.

[20] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2015.

[21] W. Guo, D. Mu, J. Xu, P. Su, G. Wang, and X. Xing, "LEMNA: Explaining Deep Learning Based Security Applications," in *Proceedings of ACM SAC Conference on Computer and Communications (CCS)*, 2018.

[22] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[24] W. He, J. Wei, X. Chen, N. Carlini, and D. Song, "Adversarial Example Defenses: Ensembles of Weak Defenses are not Strong," in *USENIX Workshop on Offensive Technologies*, 2017.

[25] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[26] A. Karpathy, J. Johnson, and L. Fei-Fei, "Visualizing and Understanding Recurrent Networks," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2016.

[27] B. Kepes, "eBrevia Applies Machine Learning to Contract Review," https://www.forbes.com/, 2015.

[28] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial Machine Learning at Scale," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2017.

[29] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[30] M. Lin, Q. Chen, and S. Yan, "Network in Network," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2014.

[31] X. Ling, S. Ji, J. Zou, J. Wang, C. Wu, B. Li, and T. Wang, "DEEPSEC: A Uniform Platform for Security Analysis of Deep Learning Model," in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, 2019.

[32] N. Liu, H. Yang, and X. Hu, "Adversarial Detection with Model Interpretation," in *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, 2018.

[33] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into Transferable Adversarial Examples and Black-Box Attacks," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2017.

[34] D. Lowd and C. Meek, "Adversarial Learning," 2005.

[35] X. Ma, B. Li, Y. Wang, S. M. Erfani, S. Wijewickrema, G. Schoenebeck, D. Song, M. E. Houle, and J. Bailey, "Characterizing Adversarial Subspaces Using Local Intrinsic Dimensionality," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.

[36] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.

[37] B. Marr, "First FDA Approval For Clinical Cloud-Based Deep Learning In Healthcare," https://www.forbes.com/, 2017.

[38] D. Meng and H. Chen, "MagNet: A Two-Pronged Defense Against Adversarial Examples," in *Proceedings of ACM SAC Conference on Computer and Communications (CCS)*, 2017.

[39] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal Adversarial Perturbations," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[40] L. Muñoz González, B. Biggio, A. Demontis, A. Paudice, V. Wongrassamee, E. C. Lupu, and F. Roli, "Towards Poisoning of Deep Learning Algorithms with Back-gradient Optimization," in *Proceedings of ACM Workshop on Artificial Intelligence and Security (AISec)*, 2017.

[41] W. J. Murdoch, P. J. Liu, and B. Yu, "Beyond Word Importance: Contextual Decomposition to Extract Interactions from LSTMs," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.

[42] W. J. Murdoch and A. Szlam, "Automatic Rule Extraction from Long Short Term Memory Networks," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2017.

[43] B. Nelson, B. I. P. Rubinstein, L. Huang, A. D. Joseph, S. J. Lee, S. Rao, and J. D. Tygar, "Query Strategies for Evading Convex-Inducing Classifiers," *J. Mach. Learn. Res.*, vol. 13, pp. 1293–1332, 2012.

[44] A. Nguyen, J. Yosinski, and J. Clune, "Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[45] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in Machine Learning: from Phenomena to Black-box Attacks Using Adversarial Samples," *ArXiv e-prints*, 2016.
[46] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks," in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, 2016.
[47] N. Papernot, P. D. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The Limitations of Deep Learning in Adversarial Settings," in *Proceedings of IEEE European Symposium on Security and Privacy (Euro S&P)*, 2016.
[48] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why Should I Trust You?": Explaining the Predictions of Any Classifier," in *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, 2016.
[49] O. Ronneberger, P.Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.
[50] S. Sabour, N. Frosst, and G. E Hinton, "Dynamic Routing Between Capsules," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2017.
[51] A. Satariano, "AI Trader? Tech Vet Launches Hedge Fund Run by Artificial Intelligence," http://www.dailyherald.com/, 2017.
[52] F. Scherer, "Heinrich von Stackelberg's Marktform und Gleichgewicht," *Journal of Economic Studies*, vol. 23, no. 5/6, pp. 58–70, 1996.
[53] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017.
[54] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning Important Features Through Propagating Activation Differences," in *Proceedings of IEEE Conference on Machine Learning (ICML)*, 2017.
[55] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the Game of Go with Deep Neural Networks and Tree Search," *Nature*, no. 7587, pp. 484–489, 2016.
[56] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2014.
[57] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2014.
[58] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "SmoothGrad: Removing Noise by Adding Noise," in *International Conference on Machine Learning Workshop*, 2017.
[59] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank," in *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.
[60] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for Simplicity: The All Convolutional Net," *Proceedings of International Conference on Learning Representations (ICLR)*, 2015.
[61] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic Attribution for Deep Networks," in *Proceedings of IEEE Conference on Machine Learning (ICML)*, 2017.
[62] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2014.
[63] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
[64] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing Properties of Neural Networks," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2014.
[65] G. Tao, S. Ma, Y. Liu, and X. Zhang, "Attacks Meet Interpretability: Attribute-Steered Detection of Adversarial Samples," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2018.
[66] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble Adversarial Training: Attacks and Defenses," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.
[67] W. Wang, N. Yang, F. Wei, B. Chang, and M. Zhou, "Gated Self-Matching Networks for Reading Comprehension and Question Answering," in *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017.
[68] H. Xiao, B. Biggio, G. Brown, G. Fumera, C. Eckert, and F. Roli, "Is Feature Selection Secure against Training Data Poisoning?" in *Proceedings of IEEE Conference on Machine Learning (ICML)*, 2015.
[69] W. Xu, D. Evans, and Y. Qi, "Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks," in *Proceedings of Network and Distributed System Security Symposium (NDSS)*, 2018.
[70] Q. Zhang, Y. Nian Wu, and S.-C. Zhu, "Interpretable Convolutional Neural Networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
[71] Q. Zhang, Y. N. Wu, and S.-C. Zhu, "Interpretable Convolutional Neural Networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
[72] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning Deep Features for Discriminative Localization," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

APPENDIX

A. ACID Attack against GRAD-CAM

Gradient-weighted Class Activation Mapping (GRAD-CAM) [53] is another feature-guided interpretation model, which generalizes CAM. Similar to CAM, it also formulates the class scores as weighted summation of the feature maps of the last convolutional layer. Differently, it projects global averaged gradients back to the convolutional feature maps:

$$w_{k,y} = \frac{1}{Z} \sum_{i,j} \frac{\partial \phi_y}{\partial a_k[i,j]} \tag{15}$$

where Z is the normalization constant. Then the attribution map is defined as:

$$m_y[i,j] = \mathrm{ReLU}\left(\sum_k w_{k,y}\, a_k[i,j] \right) \tag{16}$$

To attack GRAD-CAM, we consider the following optimization formulation:

$$\begin{array}{ll} \min_r & \ell_{\text{adv}}(x_0 + r, t) + \lambda \|m_0 - m_*\| \\ \text{s.t.} & \|r\| \le \epsilon. \end{array} \tag{17}$$

Note that though the gradient of $w_{k,y}(x)$ with respect to x is zero almost everywhere, it is feasible to find high-quality solutions to Eq. (17) with stochastic gradient descent methods, since a_k has non-zero gradients with respect to x.

	Successful Rate	ℓ_p $(p=1)$	ℓ_p $(p=2)$
PGD	98.00% (0.99)	0.25 ± 0.06	0.31 ± 0.06
ACID	92.57% (0.98)	0.12 ± 0.04	0.15 ± 0.05

Table XI. Attack effectiveness of ACID against GRADCAM.

B. Optimized MASK Formulation

The complete optimization objective for MASK in [18] is given as follows:

$$\begin{array}{ll} \min\limits_m & \lambda_1 \|\nabla m\| + \lambda_2 \|1 - m\|_1 + \mathbb{E}_\tau \left[f_y \left(\phi(x(\cdot - \tau), m) \right) \right] \\ \text{s.t.} & 0 \le m \le 1 \end{array} \tag{18}$$

Here the term $r_{\text{tv}}(m)$ is the total variation of m, which reduces its noise and artifacts; the term $\|1-m\|_1$ encourages the sparsity of m; $\phi(x;m)$ is the perturbation operator which blends x with Gaussian noise (controlled by the parameter τ); and λ_1 and λ_2 are the regularized coefficients for total variation and sparsity respectively.

C. Analysis of ACID Attack against Perturbation-guided Interpretation

We conduct an analysis of ACID attack against MASK.

Proposition 2. *Let $f(x,y):\mathbb{R}^m\times\mathbb{R}^n\to\mathbb{R}$ be a function that is second-order continuous and has partial derivatives in the neighborhood $\mathcal{N}=\mathcal{N}_x\times\mathcal{N}_y$ of a point (x_0,y_0). If it holds that,*

- ***A1.*** *for each $x\in\mathcal{N}_x$, there is a unique $g(x)\overset{def}{=}y\in\mathcal{N}_y$ such that (x,y) is a local minimizer of $f(x,\cdot)$;*
- ***A2.*** *y_0 is the unique local minimizer of $f(x_0,\cdot)$ for $y\in\mathcal{N}_y$;*
- ***A3.*** *Hessian of $f(x_0,\cdot), H(x_0,\cdot)=\nabla_y^2 f(x_0,y_0)$ is non-degenerate at y_0; in other word $\det(H(x_0,y_0))>0$.*

Then for every $g_0\in\mathcal{N}_y$, the gradient of $G(x)=\frac{1}{2}\|g(x)-g_0\|_2^2$ at $x=x_0$ is:

$$-\nabla_{xy}f(x_0,y_0)(\nabla_y^2 f(x_0,y_0))^{-1}(g(x_0)-g_0) \tag{19}$$

where $\nabla_{xy}f(x,y)$ is an $m\times n$ matrix whose entries are $m_{i,j}=\frac{\partial^2 f}{\partial x_i\partial y_j}(x,y)$.

Proof. Let $Q:\mathbb{R}^m\times\mathbb{R}^n\to\mathbb{R}^n$ be the partial derivative function of f with respec to y, i.e. $Q(x,y)=\nabla_y f(x,y)$. Then Eq. (19) is equivalent to

$$-\nabla_{xy}Q(x_0,y_0)(\nabla_y Q(x_0,y_0))^{-1}(g(x_0)-g_0) \tag{20}$$

Since y_0 is a local minimum of $f(x_0,\cdot)$, it holds that $Q(x_0,y_0)=0$. Based on the last assumption, for $J=\nabla_y Q(x_0,y_0)=\nabla_y^2 f(x_0,y_0)$, $\det(J)\neq 0$. According to the *implicit function* theorem, there exists a neighborhood of x_0, $\hat{\mathcal{N}}_x\subset\mathcal{N}_x\subset\mathbb{R}^m$, a neighborhood of y_0 $\hat{\mathcal{N}}_y\subset\mathcal{N}_y\subset\mathbb{R}^n$, and a unique smooth function $g(x):\hat{\mathcal{N}}_x\to\hat{\mathcal{N}}_y$ such that

$$Q(x,g(x))=0\quad\forall x\in\hat{\mathcal{N}}_x \tag{21}$$

Since for each $x\in\mathcal{N}_x$, $f(x,y)$ has a unique local minimum near y_0, then the local minimizer is $g(x)$ due to the first-order optimal condition. Computing the gradient with respect to x for Eq. (21), we get:

$$\nabla_x g(x_0)=-\nabla_x Q(x_0,y_0)(\nabla_y Q(x_0,y_0))^{-1} \tag{22}$$

Finally, we arrive at Eq. (20) with the product rule for the gradient. □

Back to the MASK case, let $\ell(x,m;y)$ denote the mask loss in Eq. (18) for x, an attribution map m with target label y, and target map m_0. Though $\ell(x,m;y)$ is not even continuously differentiable, we can relax it for analysis purpose to gain some insights. Therefore, we assume $l(\cdot,\cdot;y)$ satisfies the assumptions of $f(.,.)$ in Prop 2. At the t-th iteration, if the optimal attribution map of the current input x_t is m_t^*, then we can take a step towards the direction of

$$\Delta_t=-\nabla_{xm}l(x_t,m_t^*;y)(\nabla_m^2\ell(x_t,m_t^*;y))^{-1}(m_t^*-m_0) \tag{23}$$

to make the attribution map of x_{t+1} closer to m_0. While in practice we only have m_t, a noisy estimate of m_t^*, we plug m_t into Eq. (23), and let $H=\nabla_m^2\ell(x_t,m_t;y)$ denote the Hessian matrix of $\ell(x_t,\cdot;y)$ with fixed x. For some step size $\alpha>0$:

$$\begin{aligned}\alpha\Delta_t\approx&-\alpha\nabla_{xm}\ell(x_t,m_t;y)(\nabla_m^2\ell(x_t,m_t;y))^{-1}(m_t-m_0)\\=&-\alpha\nabla_{xm}\ell(x_t,m_t;y)H^{-1}(m_t-m_0)\\=&\nabla_x\left(-\alpha H^{-1}\nabla_m\ell(x_t,m_t;y)\right)(m_t-m_0)\\=&\nabla_x\underbrace{\left(m_t-\alpha H^{-1}\nabla_m\ell(x_t,m_t;y)\right)}_{\text{inner update step}}(m_t-m_0)\end{aligned} \tag{24}$$

In the context of optimization, the inner update step of Eq. (24) is a standard iteration of Newton's method. In our attack against MASK, we replace the Newton's step with an Adam step to get faster convergence and to avoid the problem of vanishing Hessian of a ReLU network.

D. More Experimental Results

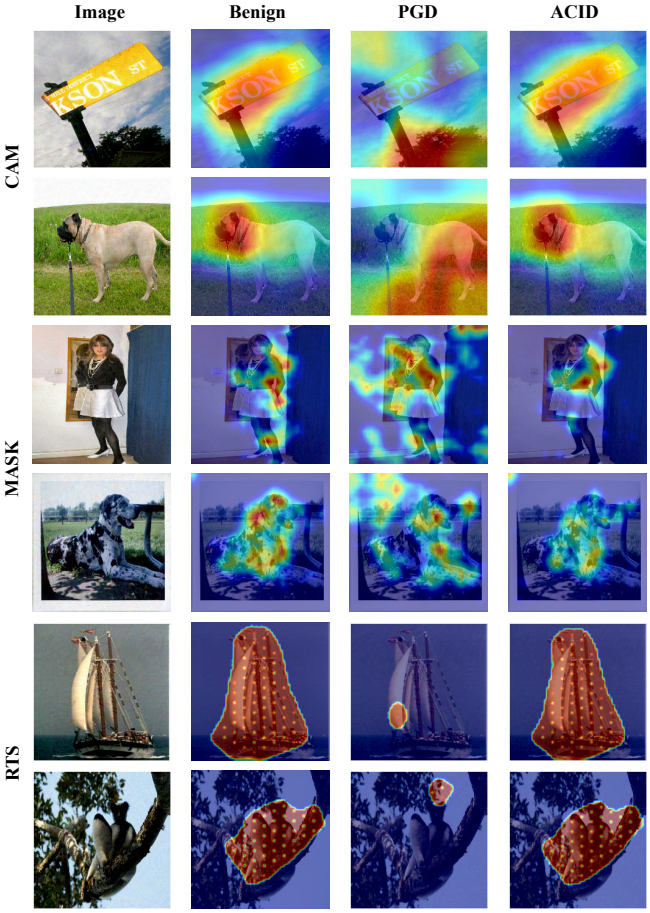

Figure 10: Visualization of attribution maps of benign and adversarial (by the PGD and ACID attacks) inputs with respect to CAM, MASK, and RTS on DenseNet.

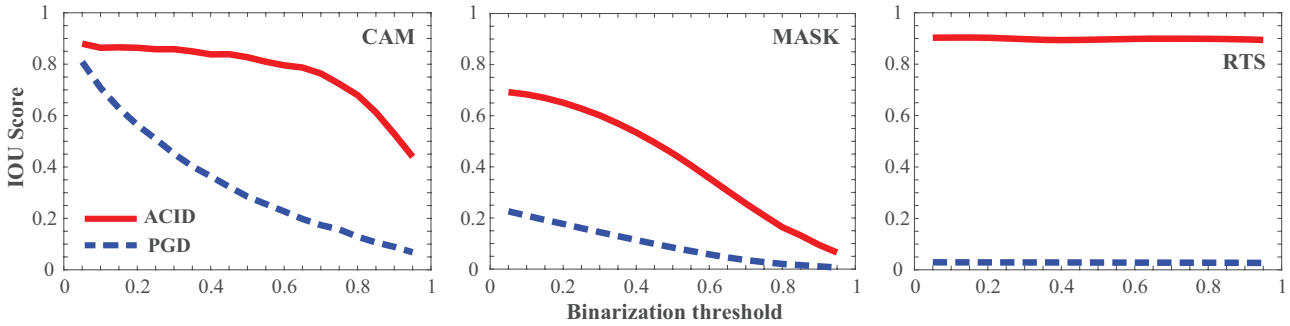

Figure 11: IOU scores of adversarial attribution maps (by PGD and ACID) with respect to benign maps on DenseNet.