# Adversarial Attacks and Defenses: An Interpretation Perspective

Ninghao Liu†, Mengnan Du†, Ruocheng Guo‡, Huan Liu‡, Xia Hu†
Department of Computer Science and Engineering, Texas A&M University, TX, USA
Computer Science & Engineering, Arizona State University, Tempe, AZ, USA
†{nhliu43, dumengnan, xiahu}@tamu.edu, ‡{rguo12, huanliu}@asu.edu

## ABSTRACT

Despite the recent advances in a wide spectrum of applications, machine learning models, especially deep neural networks, have been shown to be vulnerable to *adversarial attacks*. Attackers add carefully-crafted perturbations to input, where the perturbations are almost imperceptible to humans, but can cause models to make wrong predictions. Techniques to protect models against adversarial input are called *adversarial defense* methods. Although many approaches have been proposed to study adversarial attacks and defenses in different scenarios, an intriguing and crucial challenge remains that how to really understand model vulnerability? Inspired by the saying that "if you know yourself and your enemy, you need not fear the battles", we may tackle the challenge above after interpreting machine learning models to open the black-boxes. The goal of *model interpretation*, or *interpretable machine learning*, is to extract human-understandable terms for the working mechanism of models. Recently, some approaches start incorporating interpretation into the exploration of adversarial attacks and defenses. Meanwhile, we also observe that many existing methods of adversarial attacks and defenses, although not explicitly claimed, can be understood from the perspective of interpretation. In this paper, we review recent work on adversarial attacks and defenses, particularly from the perspective of machine learning interpretation. We categorize interpretation into two types, feature-level interpretation, and model-level interpretation. For each type of interpretation, we elaborate on how it could be used for adversarial attacks and defenses. We then briefly illustrate additional correlations between interpretation and adversaries. Finally, we discuss the challenges and future directions for tackling adversary issues with interpretation.

## Keywords

Adversarial attacks, adversarial defenses, interpretation, explainability, deep learning

## 1. INTRODUCTION

Machine learning (ML) techniques, especially recent deep learning models, are progressing rapidly and have been increasingly applied in various applications. Nevertheless, concerns have been posed about the security and reliability issues of ML models. In particular, many deep models are susceptible to adversarial attacks [1; 2]. That is, after adding certain well-designed but human imperceptible perturbation or transformation to a clean data instance, we are able to manipulate the prediction of the model. The data instances after being attacked are called *adversarial samples*. The phenomenon is intriguing since clean samples and adversarial samples are usually not distinguishable to humans. Adversarial samples may be predicted dramatically differently from clean samples, but the predictions usually do not make sense to a human.

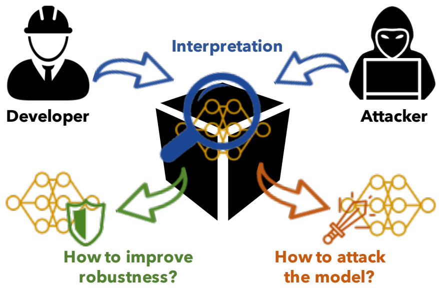


Figure 1: Interpretation can either provide directions for improving model robustness or attacking on its weakness.

The model vulnerability to adversarial attacks has been discovered in various applications or under different constraints. For examples, approaches for crafting adversarial samples have been proposed in tasks such as classification (e.g., on image data [3], text data [4], tabular data [5], graph data [6; 7]), object detection [8], and fraud detection [9]. Adversarial attacks could be initiated under different constraints, such as assuming limited knowledge of attackers on target models [10; 11], assuming higher generalization level of attack [12; 13], posing different real-world constraints on attack [14; 15]. Given the advances, several questions could be posted. First, are these advances relatively independent of each other, or is there an underlying perspective from which we can discover the commonality behind them? Second, should adversarial samples be seen as the negligent corner cases that could be fixed by putting patches to models, or are they deeply rooted in the internal working mechanism of models that it is not easy to get rid of?

Motivated by the idiom that "if you know yourself and your enemy, you need not fear the battles" from *The Art of War*, in this paper, we answer the above questions and review the recent advances of adversarial attack and defense approaches from the perspective of interpretable machine learning. The relation between model interpretation and model robustness is illustrated in Figure 1. On the one hand, if adversaries know how the target model works, they may utilize it to find model weakness and initiate attacks accordingly. On the other hand, if model developers know how the model works, they could identify the vulnerability and work on remediation in advance. Interpretation refers to the human-understandable information explaining what a model has learned or how a model makes predictions. Exploration of model interpretability has attracted many interests in recent years, because recent machine learning techniques, especially deep learning models, have been criticized due to lack of transparency. Some recent work starts to involve interpretability in the analysis of adversarial robustness. Also, although not being explicitly specified, in this survey, we will show that many existing adversary-related work can be comprehended from another perspective as an extension of model interpretation.

Before connecting the two domains, we first briefly introduce the subjects of interpretation to be covered in this paper. *Interpretability* is defined as "the ability to explain or to present in understandable terms to a human [16]". Although a formal definition of interpretation still remains elusive [16; 17; 18; 19], the overall goal is to obtain and transform information from models or their behaviors into a domain that human can make sense of [20]. For a more structured analysis, we categorize existing work into two categories: feature-level interpretation and model-level interpretation, as shown in Figure 2. Feature-level interpretation targets to find the most important features in a data sample for its prediction. Model-level interpretation explores the functionality of model components, and their internal states after being fed with input. This categorization is based on whether the internal working mechanism of models is involved in interpretation.

Following the above categorization, the overall structure of this article is organized as below. To begin with, we briefly introduce different types of adversarial attack and defense strategies in Section 2. Then, we introduce different categories of interpretation approaches, and demonstrate in detail how interpretation correlates to the attack and defense strategies. Specifically, we discuss feature-level interpretation in Section 3 and model-level interpretation in Section 4. After that, we extend the discussion to additional relations between interpretation and adversarial aspects of models in Section 5. Finally, we discuss some opening challenges for future work in Section 6.

## 2. ADVERSARIAL MACHINE LEARNING

Before understanding how interpretation helps adversarial attack and defense, we first provide an overview of existing attack and defense methodologies.

### 2.1 Adversarial Attacks

In this subsection, we introduce different types of threat models for adversarial attacks. The overall threat models may be categorized under different criteria. Based on different application scenarios, conditions, and adversary capabilities, specific attack strategies will be deployed.

#### 2.1.1 Untargeted vs Targeted Attack

Based on the goal of attackers, the threat models can be classified into targeted and untargeted ones. For *targeted* attack,

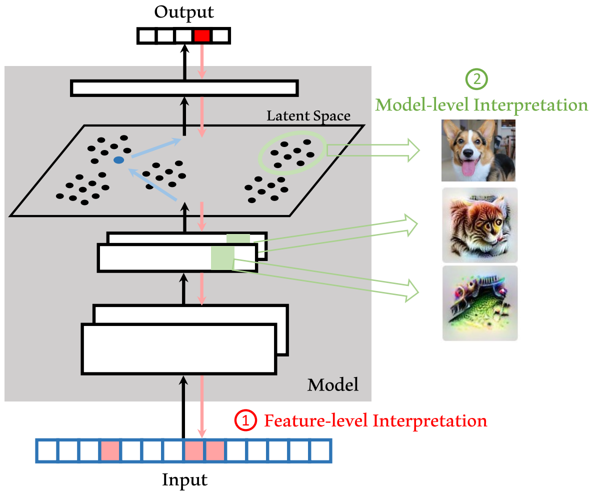


Figure 2: Illustration of feature-level interpretation and model-level interpretation for a deep model.

it attempts to mislead a model's prediction to a specific class given an instance. Let $f$ denote the target model exposed to adversarial attack. A clean data instance is $\mathbf{x}_0 \in X$, and $X$ is the input space. We consider classification tasks, so $f(\mathbf{x}_0) = c, c \in \{1, 2, ..., C\}$. One way of formulating the task of targeted attack is as below [2]:

$$\min_{\mathbf{x} \in X} d(\mathbf{x}, \mathbf{x}_0), \quad \text{s.t. } f(\mathbf{x}) = c' \tag{1}$$

where $c' \neq c$, and $d(\mathbf{x}, \mathbf{x}_0)$ measures the distance between the two instances. A typical choice of distance measure is $l_p$ norms, where $d(\mathbf{x}, \mathbf{x}_0) = \|\mathbf{x} - \mathbf{x}_0\|_p$. The core idea is to add small perturbation to the original instance $\mathbf{x}_0$ to make it being classified as $c'$. However, in some cases, it is important to increase the confidence of perturbed samples being misclassified, so the task may also be formulated as:

$$\max_{\mathbf{x} \in X} f_{c'}(\mathbf{x}), \quad \text{s.t. } d(\mathbf{x}, \mathbf{x}_0) \leq \delta \tag{2}$$

where $f_{c'}(\mathbf{x})$ denotes the probability or confidence that $\mathbf{x}$ is classified as $c'$ by $f$, and $\delta$ is a threshold limiting perturbation magnitude. For *untargeted* attack, its goal is to prevent a model from assigning a specific label to an instance. The objective of untargeted attack could be formulated in a similar way as targeted attack, where we just need to change the constraint as $f(\mathbf{x}) \neq c$ in Equation 1, or change the objective as $\min_{\mathbf{x} \in X} f_c(\mathbf{x})$ in Equation 2.

In some scenarios, the two types of attacks above are also called *false positive* attack and *false negative* attack. The former aims to make models misclassify negative instances as positive, while the latter tries to mislead models to classify positive instances as negative. False positive attacks and false negative attacks sometimes are also called Type-I attacks and Type-II attacks, respectively.

#### 2.1.2 One-Shot vs Iterative Attack

According to practical constraints, adversaries may initiate one-shot or iterative attacks to target models. In *one-shot* attack, they have only one chance to generate adversarial samples, while *iterative attack* could take multiple steps to find the better perturbation direction. Iterative attacks can generate more effective adversarial samples than one-shot attacks. However, it also requires more queries to the target model and more computation to initiate each attack, which may limit its application in some computational-intensive tasks.

#### 2.1.3 Data-Dependent vs Universal Attack

According to information sources, adversarial attacks could be data-dependent or data-independent. In *data dependent* attack, perturbations are customized based on the target instance. For example, in Equation 1, the adversarial sample $\mathbf{x}$ is crafted based on the original instance $\mathbf{x}_0$. However, it is also possible to generate adversarial samples without referring to the input instance, and it is also named as *universal* attack [12; 21]. The problem can be abstracted as looking for a perturbation vector $\mathbf{v}$ so that

$$f(\mathbf{x}+\mathbf{v}) \neq f(\mathbf{x}) \text{ for "most" } \mathbf{x} \in X. \tag{3}$$

We may need a number of training samples to obtain $\mathbf{v}$, but it does not rely on any specific input at test time. Adversarial attacks can be implemented efficiently once the vector $\mathbf{v}$ is solved.

#### 2.1.4 Perturbation vs Replacement Attack

Adversarial attacks can also be categorized based on the way of input distortion. In *perturbation* attack, input features are shifted by specific noises so that the input is misclassified by the model. In this case, let $\mathbf{x}^*$ denote the final adversarial sample, then it can be obtained via

$$\mathbf{x}^* = \mathbf{x}_0 + \Delta\mathbf{x}, \tag{4}$$

and usually $\|\Delta\mathbf{x}\|_p$ is small.
In *replacement* attack, certain parts of the input are replaced by adversarial patterns. Replacement attack is more natural in physical scenarios. For example, criminals may want to wear specifically designed glasses to prevent them from being recognized by computer vision systems [1]. Also, surveillance cameras may fail to detect persons wearing clothes attached with adversarial patches [14]. Suppose $\mathbf{v}$ denotes the adversarial pattern, then replacement attack can be represented by using a mask $\mathbf{m} \in \{0,1\}^{|\mathbf{x}_0|}$, so that

$$\mathbf{x}^* = \mathbf{x}_0 \odot (\mathbf{1}-\mathbf{m}) + \mathbf{v} \odot \mathbf{m} \tag{5}$$

where the symbol $\odot$ denotes element-wise multiplication.

#### 2.1.5 White-Box vs Black-Box Attack

In *white-box* attack, it is assumed that attackers know everything about the target model, which may include model architecture, weights, hyper-parameters, and even training data. White-box attacks help to discover intrinsic vulnerabilities of the target model. It works in ideal cases representing the worst scenario that defenders have to confront. *Black-box* attack assumes that attackers are only accessible to the model output, just like regular end-users. This is a more practical assumption in real-world scenarios. Although a lot of detailed information about models is occluded, black-box attacks still pose a significant threat to machine learning systems due to the transferability property of adversarial samples discovered in [11]. In this sense, an attacker could build a new model $f'$ to approximate the target model $f$, and adversarial samples created on $f'$ could still be effective to $f$.

[1]https://www.inovex.de/blog/machine-perception-face-recognition/

### 2.2 Defenses Against Adversarial Attacks

In this subsection, we briefly introduce the basic idea of different defense strategies against adversaries.

#### 2.2.1 Input Denoising

As adversarial perturbation is a type of human-imperceptible noise added to data, then a natural defense solution is to filter it out, or to use additional random transformation to offset adversarial noise. It is worth noting that $f_m$ could be added prior to model input layer [22; 23; 24], or as an internal component inside the target model [25]. Formally, for the former case, given an instance $\mathbf{x}^*$ which is probably affected by adversaries, we hope to design a mapping $f_m$, so that $f(f_m(\mathbf{x}^*)) = f(\mathbf{x}_0)$. For the latter case, the idea is similar except that $f$ is replaced by certain intermediate layer output $h$.

#### 2.2.2 Model Robustification

Refining the model to prepare itself against a potential threat from adversaries is another widely applied strategy. The model refinement could be achieved from two directions: changing the training objective, or modifying the model structure. Some examples of the former one include adversarial training [2; 1], and replacing empirical training loss with robust training loss [26]. The intuition behind it is to consider in advance the threat of adversarial samples during model training, so that the resultant model gains robustness from training. Examples of model modification include model distillation [27], applying layer discretization [28], controlling neuron activations [29]. Formally, let $f'$ denote the robust model, the goal is to make $f'(\mathbf{x}^*)) = f'(\mathbf{x}_0) = y$.

#### 2.2.3 Adversarial Detection

Unlike the previous two strategies where we hope to discover the true label given an instance, adversarial detection tries to identify whether the given instance is polluted by adversarial perturbation. The general idea is to build another predictor $f_d$, so that $f_d(\mathbf{x}) = 1$ if $\mathbf{x}$ has been polluted, and otherwise $f_d(\mathbf{x}) = 0$. The establishment process of $f_d$ could follow the normal routine of building a binary classifier [30; 31; 32].
Input denoising and model robustification methods proactively recover the prediction from influences of adversarial attacks by focusing on modifying the input data and model architectures, respectively. Adversarial detection methods passively decide whether the model should make predictions against the input in order not to be fooled. Implementations of proactive strategies are usually more challenging than passive ones.

## 3. FEATURE-LEVEL INTERPRETATION IN ADVERSARIAL MACHINE LEARNING

Feature-level interpretation is a widely used post-hoc method to identify feature importance for a prediction result. It focuses on the end-to-end relation between input and output, instead of carefully examining the internal states of models. Some examples include measuring the importance of phrases of sentences in text classification [33], and pixels in image classification [34]. In this section, we will discuss how

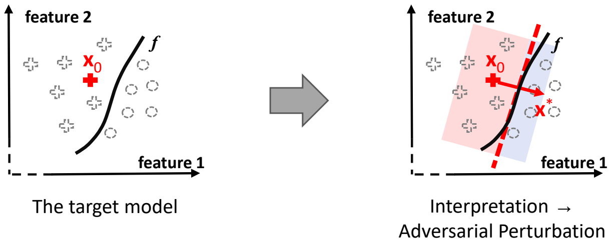


Figure 3: Interpretation naturally unveils the direction of adversarial perturbation ($g$ denotes the local interpreter).

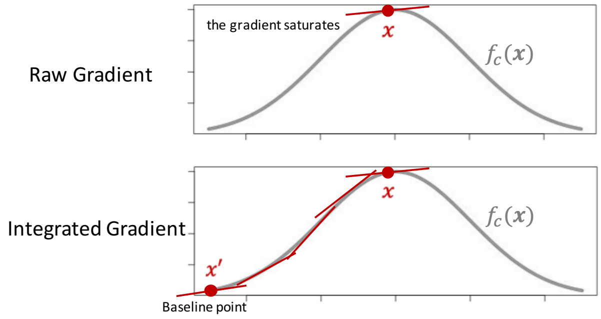


Figure 4: Raw gradients only consider the local sensitivity of output to input value changes, which could be limited in measuring the contribution of a feature to the prediction.

this type of interpretation correlates with the attack and defense of adversaries, given that many works on adversarial machine learning do not analyze adversaries from this perspective.

## 3.1 Feature-Level Interpretation for Understanding Adversarial Attacks

In this part, we will show that many feature-level interpretation techniques are closely coupled with existing adversarial attack methods, thus providing another perspective to understand adversarial attacks.

### *3.1.1 Gradient-Based Techniques*

Following the notations in previous discussion, we let $f_c(\mathbf{x}_0)$ denote the probability that model $f$ classifies the input instance $\mathbf{x}_0$ as class $c$. One of the intuitive ways to understand why such prediction is derived is to attribute prediction $f_c(\mathbf{x}_0)$ to feature importance in $\mathbf{x}_0$. According to [35], $f_c(\mathbf{x}_0)$ can be approximated with a linear function surrounding $\mathbf{x}_0$ by computing the first-order Taylor expansion:

$$f_c(\mathbf{x}) \approx f_c(\mathbf{x}_0) + \mathbf{w}_c^T \cdot (\mathbf{x} - \mathbf{x}_0) \tag{6}$$

where $\mathbf{w}_c$ is the gradient of $f_c$ with respect to input at $\mathbf{x}_0$, i.e., $\mathbf{w}_c = \nabla_{\mathbf{x}} f_c(\mathbf{x}_0)$. From the interpretation perspective, $\mathbf{w}_c$ entries of large magnitude correspond to the features that are important around the current output.

However, another perspective to comprehend the above equation is that, the interpretation $\mathbf{w}_c$ also indicates the most effective direction to change the prediction result by perturbing input away from $\mathbf{x}_0$. If we let $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}_0 \propto -\mathbf{w}_c$, we are attacking the model $f$ with respect to the input-label pair $(\mathbf{x}_0, c)$. Such perturbation method is closely related to the Fast Gradient Sign (FGS) attacking method [1], where:

$$\Delta\mathbf{x} = \epsilon \cdot sign(\nabla_{\mathbf{x}} J(f, \mathbf{x}_0, c)), \tag{7}$$

except that (1) FGS computes the gradient of a certain cost function $J$ nested outside $f$, and (2) it applies an additional $sign()$ operation on gradient for processing images. However, if we define $J$ with cross entropy loss, and the true label of $\mathbf{x}_0$ is $c$, then

$$\nabla_{\mathbf{x}} J(f, \mathbf{x}_0, c) = -\nabla_{\mathbf{x}} \log f_c(\mathbf{x}_0) = -\frac{1}{f_c(\mathbf{x}_0)} \nabla_{\mathbf{x}} f_c(\mathbf{x}_0), \tag{8}$$

which points to exactly the opposite direction of interpretation $\mathbf{w}_c$. The high-level idea behind this case is that, if the interpretation of a model is known, a straightforward way to undermine the model is to remove the important information or components relevant to the interpretation.

The traditional FGS method is proposed under untargeted attacks, where the goal is to impede input from being correctly classified. For targeted attack, where the goal is to misguide the model prediction towards a specific class, a typical way is Box-constrained L-BFGS (L-BFGS-B) method [2]. Assume $c'$ is the target label, the problem of L-BFGS-B is formulated as:

$$\underset{\mathbf{x} \in X}{\operatorname{argmin}} \quad \alpha \cdot d(\mathbf{x}, \mathbf{x}_0) + J(f, \mathbf{x}, c') \tag{9}$$

where $d$ is considered to control the perturbation distance, and $X$ is the input domain (e.g., $[0, 255]$ for each channel of image input). The goal of attack is to make $f(\mathbf{x}) = c'$, while making $d(\mathbf{x}, \mathbf{x}_0)$ to be small. Suppose we apply gradient descent to solve the problem, and $\mathbf{x}_0$ is the starting point. Similar to the previous discussion, if we define $J$ as the cross entropy loss, then

$$-\nabla_{\mathbf{x}} J(f, \mathbf{x}_0, c')) = \nabla_{\mathbf{x}} \log f_{c'}(\mathbf{x}_0) \propto \mathbf{w}_{c'} \, . \tag{10}$$

On one hand, $\mathbf{w}_{c'}$ locally and linearly interprets $f_{c'}(\mathbf{x}_0)$, and it also serves the most effective direction to make $\mathbf{x}_0$ towards being classified as $c'$.

According to the taxonomy of adversarial attacks, the two scenarios discussed above can also be categorized into: (1) one-shot attack, since we only perform interpretation once, (2) data-dependent attack, since the perturbation direction is related with $\mathbf{x}_0$, (3) white-box attack, since model gradients are available. Other types of attack could be crafted if different interpretation strategies are applied, which will be discussed in later sections.

**Improved Gradient-Based Techniques.** The interpretation methods based on raw gradients, as discussed above, are usually unstable and noisy [36; 37]. The possible reasons include: (1) the target model's prediction function itself is not stable; (2) gradients only consider the local output-input relation so that its scope is too limited (Figure 4); (3) the prediction mechanism is too complex to be approximated by a linear substitute. Some approaches for improving interpretation (i.e., potential adversarial attack) are as below.

- **Region-Based Exploration:** To reduce random noises in interpretation, SmoothGrad is proposed in [38], where the final interpretation $\mathbf{w}_c$, as a sensitivity map, is obtained by averaging multiple interpretation results of instances sampled around the target instance $\mathbf{x}_0$, i.e., $\mathbf{w}_c = \sum_{\mathbf{x}' \in \mathcal{N}(\mathbf{x}_0)} \frac{1}{|\mathcal{N}(\mathbf{x}_0)|} \nabla f_c(\mathbf{x}')$. The averaged sensitivity map will be visually sharpened. A straightforward way to extend it for adversarial attack is to perturb input by reversing the averaged interpretation. Furthermore, [39] designed a different strategy by adding a step of random perturbation before gradient computation in attack, to jump out of the non-smooth vicinity of the initial instance. Spatial averaging is a common technique to stabilize output. For example, [40] applies it as a defense method to derive more stable model predictions.

- **Path-Based Integration:** To improve interpretation and consider a broader input scope, [41] proposes Integrated Gradient (InteGrad). After setting a baseline point $\mathbf{x}^b$, e.g., an all-black image in classification tasks, the interpretation is defined as:

$$\mathbf{s}_c = \frac{(\mathbf{x}_0 - \mathbf{x}^b)}{D} \circ \sum_{d=1}^{D} [\nabla f_c](\mathbf{x}^b + \frac{d}{D}(\mathbf{x}_0 - \mathbf{x}^b)), \tag{11}$$

which is the weighted sum of gradients along the straight-line path from $\mathbf{x}_0$ to the baseline point $\mathbf{x}^b$. Let $\mathbf{s}_c(m)$ denote the $m$-th entry of $\mathbf{s}_c$, then the prediction function could be decomposed as below:

$$f_c(\mathbf{x}) \approx f_c(\mathbf{x}^b) + \sum_{m=1}^{M} \mathbf{s}_c(m), \tag{12}$$

which is different from the decomposition in Eq 6. Here $\mathbf{s}_c(m)$ denotes the contribution of the $m$-th feature to the prediction result. Therefore, a new type of adversarial attack could be conducted by deleting or removing those features with high contribution scores. This type of feature deletion or feature occlusion attack is different from FGS that perturbs feature values.

Interestingly, although in many cases gradient-based interpretation is intuitive as visualization to show that the model is functioning well, it may be an illusion since we can easily transform interpretation into adversarial perturbation.

#### 3.1.2 Distillation-Based Techniques

The interpretation techniques discussed so far require gradient information $\nabla_{\mathbf{x}} f$ from models. Meanwhile, it is possible to extract interpretation without querying a model $f$ more than $f(\mathbf{x})$. This type of interpretation method, here named as the distillation-based method, can also be used for adversarial attacks. Since no internal knowledge is required from the target model, they are usually used for black-box attacks.

The main idea of applying distillation for interpretation is to use an interpretable model $g$ (e.g., a linear model) to locally mimic the behavior of the target deep model $f$ [42; 43]. Once we obtain $g$, existing white-box attack methods could be applied to craft adversarial samples [5]. In addition, given an instance $\mathbf{x}_0$, to guarantee that $g$ more accurately mimics the behaviors of $f$, we could further require that $g$ locally approximates $f$ around the instance. The objective is thus as below:

$$\min_g \mathcal{L}(f, g, \mathbf{x}_0) + \alpha \cdot C(g), \tag{13}$$

where $\mathcal{L}$ denotes the approximation error around $\mathbf{x}_0$. For example, in LIME [44]:

$$\mathcal{L}(f, g, \mathbf{x}_0) = \sum_{\mathbf{x}' \in \mathcal{N}(\mathbf{x}_0)} \exp(-d(\mathbf{x}_0, \mathbf{x}')) \| f(\mathbf{x}') - g(\mathbf{x}')\|_2^2, \tag{14}$$

and $\mathcal{N}(\mathbf{x}_0)$ denotes the local region around $\mathbf{x}_0$. In addition, LEMNA [45] adopts mixture regression models for $g$ and fused lasso as regularization $C(g)$. After obtaining $g$, we can craft adversarial samples targeting $g$ by removing important features or perturbing input towards the reversed direction of interpretation. According to the property of transferability [11], an adversarial sample that successfully fools $g$ is also likely to fool $f$. The advantages are two-fold. First, the process is model-agnostic and does not assume availability to gradients. It could be used for black-box attacks or attacking certain types of models (such as tree-based models) that do not use gradient backpropagation in training. Second, one-shot attacks on $g$ could be more effective thanks to the smoothness term $C(g)$ as well as extending the consideration to include the neighborhood of $\mathbf{x}_0$ [46]. Thus, it has the potential to make defense methods based on obfuscated gradients [47] to be less robust. However, the disadvantage is that crafting each adversarial sample requires higher computation cost.

In certain scenarios, it may be beneficial to make adversarial patterns understandable to humans as real-world simulation when identifying model vulnerability. For example, in autonomous driving, we need to consider physically-possible patterns that could cause misjudgment of autonomous vehicles [48]. One possible approach is to encourage adversarial instances to fall into the data distribution [49], which could be implemented through a regularization term $\|\mathbf{x}_0 + \Delta\mathbf{x} - AE(\mathbf{x}_0 + \Delta\mathbf{x})\|_2^2$, where $AE(\cdot)$ denotes an autoencoder. By minimizing the normalization term, the perturbed data $\mathbf{x}_0 + \Delta\mathbf{x}$ can be well modeled by the autoencoder, which implies that it is close to the data manifold. Another strategy is to predefine a dictionary, and then make the adversarial perturbation to match one of the dictionary tokens [48], or a weighted combination of the tokens [50].

#### 3.1.3 Influence-Function Based Techniques

Instead of measuring feature importance (e.g., feature sensitivity, feature contribution) as explanations, influence functions provide a new perspective by measuring the importance of data instances. Suppose $\mathbf{x}_1$, $\mathbf{x}_2$, ..., $\mathbf{x}_N$ are the training instances, and $\theta$ denotes the model parameters. Let $\mathcal{L}(\mathbf{x}_n, \theta)$ be the loss on a single instance, and $\frac{1}{N}\sum_{n=1}^{N} \mathcal{L}(\mathbf{x}_n, \theta)$ be the overall empirical loss. The optimal parameters are given by $\hat{\theta} = \mathrm{argmin}_\theta \frac{1}{N}\sum_{n=1}^{N} \mathcal{L}(\mathbf{x}_n, \theta)$. According to [51], influence function could be used to answer several questions: (1) how model parameters $\theta$ would change if an instance $\mathbf{x}_n$ is removed, (2) how model prediction on a test point $\mathbf{x}_{test}$ would change if an instance $\mathbf{x}_n$ is removed, (3) how model prediction would change if an instance $\mathbf{x}_n$ is modified. By answering the third question, through experimental demonstration, the paper shows that after injecting perturbed data instances into the training set (i.e., data poisoning), the new model will make wrong predictions on some test points.

In explanations derived from influence functions, the fundamental unit is the data instance instead of the feature. Therefore, it seems difficult to directly utilize explanation results from influence functions to initiate adversarial attacks. However, in graph analysis, influence functions are useful in studying the importance of graph components (e.g., nodes and edges) that can be regarded as the "features" of the graph. A graph can be denoted as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{v_1, v_2, ..., v_N\}$ is the set of nodes, and $\mathcal{E}$ is the set of edges. Each edge is denoted as $(v_i, v_j) \in \mathcal{E}$. An important task in graph analysis is node embedding, where we learn an embedding vector for each node. The embedding vectors can be used in downstream tasks such as node classification and link prediction. One of the fundamental requirements for learning embeddings is that the embeddings of nodes that are connected or have similar contexts (e.g., similar neighbors) should be close to each other. By utilizing influence functions, it is possible to identify how adding or deleting an edge would change the node embeddings [52]. The addition or deletion of a small number of edges can be seen as adversarial attacks on graph data.

## 3.2 Feature-Level Interpretation for Adversarial Defenses

The feature-level interpretation could be used for defense against adversaries through adversarial training and detecting model vulnerability.

### 3.2.1 *Model Robustification With Feature-Level Interpretation*

The feature-level interpretation could help adversarial training to improve model robustness. Adversarial training [1; 3] is one of the most applied proactive countermeasures to improve the robustness of the model. Feature-level interpretation could help in crafting adversarial samples to unveil the weakness of the model. The adversarial samples are then injected into the training set for data augmentation. The overall loss function can be formulated as:

$$\min_f \mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}} [\alpha J(f(\mathbf{x}), y) + (1-\alpha)J(f(\mathbf{x}^*), y)]. \quad (15)$$

In the scenario of adversarial training, feature-level interpretation helps in preparing adversarial samples $\mathbf{x}^*$, which may refer to any method discussed in Section 3.1.1 and Section 3.1.2. Although such an attack-and-then-debugging strategy has been successfully applied in many traditional cybersecurity scenarios, one key drawback is that it tends to overfit the specific approach that is used to generate $\mathbf{x}^*$. Therefore, the adversarial training is usually conducted for multiple rounds.

To train more robust models, some optimization based methods have been proposed. [26] argues that traditional Empirical Risk Minimization (ERM) fails to yield models that are robust to adversarial instances, and proposed a min-max formulation to train robust models:

$$\min_f \mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}} [\max_{\delta\in\Delta X} J(\mathbf{x}+\delta, y)], \quad (16)$$

where $\Delta X$ denotes the set of allowed perturbations. It formally defines adversarially robust classification as a learning problem to reduce adversarial expected risk. This min-max formulation provides another perspective on adversarial training, where the inner task aims to find adversarial samples, and the outer task retrains model parameters. [39] further improves its defense performance by crafting adversarial samples from multiple sources to augment training data. [53] further identifies a trade-off between robust classification error and natural classification error, which provides a solution to reduce the negative effect on model accuracy after adversarial training.

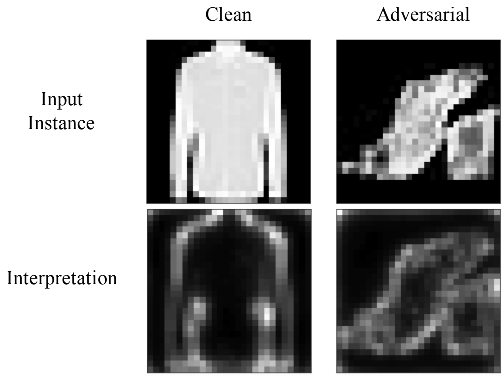


Figure 5: The interpretation of an adversarial sample may differ from the one of a clean sample. Top-left: a normal example from the shirt class of Fashion-MNIST dataset. Bottom-left: the explanation map for the classification. Top-right: an adversarial example, originally from the sandal class, that is misclassified as a shirt. Bottom-right: the explanation map for the misclassification.

Besides adversarial training, in more cases, feature-level interpretation plays the role of providing motivation for robust learning. For example, empirical interpretation results pointed out, that an intriguing property of CNN is its bias towards texture instead of shape information when making predictions [54]. To tackle this problem, [55] proposes InfoDrop, a plug-in filtering method to remove texture-intensive information during forward propagation of CNN. Feature map regions with low self-information, i.e., regions with texture patterns that contain less "surprise", tend to be filtered out. In this way, the model will pay more attention to regions such as edges and shapes, and be more robust under various scenarios including adversaries.

### 3.2.2 *Adversarial Detection With Feature-Level Interpretation*

In the scenario where a model is subject to adversarial attacks, interpretation may serve as a new type of information for directly detecting adversarial patterns. The motivation is illustrated in Figure 5. In the adversarial image which originally shows a shoe, although the model classifies it as a shirt, its interpretation result does not resemble the one obtained from the clean image of a shirt. A straightforward way to distinguish interpretations is to train another classifier as the detector trained with interpretations of both clean and adversarial instances, paired with labels indicating whether the sample is clean [56; 57; 58; 59]. Specifically, [59] directly uses gradient-based saliency map as interpretation, [58] adopts the distribution of Leave-One-Out (LOO) attribution scores, while [57] proposes a new interpretation method based on masks highlighting important regions. [60] proposes an ensemble framework called X-Ensemble for detecting adversarial samples. X-Ensemble consists of multiple sub-detectors, each of which is a convolutional neural network to classify whether an instance is adversarial or benign. The input to each sub-detector is the interpretation of the instance's prediction. More than one interpretation method is deployed, so there are multiple sub-detectors. A random forest model is then used to combine sub-detectors into a powerful ensemble detector.

In more scenarios, interpretation serves as a diagnostic tool to qualitatively identify model vulnerability. First, we could use interpretation to identify whether inputs are affected by adversarial attacks. For example, if the interpretation result shows that unreasonable evidence has been used for prediction [61], then it is possible that there exist suspicious but imperceptible input patterns. Second, interpretation may reflect whether a model is susceptible to adversarial attack. Even given a clean input instance, if the interpretation of model prediction does not make much sense to humans, then the model is at the risk of being attacked. For example, in a social spammer detection system, if the model regards certain features as important, but they are not strongly correlated with maliciousness, then attackers could easily manipulate these features without much cost to fool the system [5]. Also, in image classification, CNN models have been demonstrated to focus on local textures instead of object shapes, which could be easily utilized by attackers [54]. An interesting phenomenon in image classification is that, after refining a model with adversarial training, feature-level interpretation results indicate that the refined model will be less biased towards texture features [62].

Nevertheless, there are several challenges that impede the intuitions above from being formulated to formal defense approaches. First, the interpretation itself is also fragile in neural networks. Attackers could control prediction and interpretation simultaneously via indistinguishable perturbation [63; 64]. Second, it is difficult to quantify the robustness of a model through interpretation [36]. Manual inspection of interpretation helps discover defects in model, but visually acceptable interpretation does not guarantee model robustness. That is, defects in feature-level interpretation indicate the presence but not the absence of vulnerability.

# 4. MODEL-LEVEL INTERPRETATION IN ADVERSARIAL MACHINE LEARNING

In this review, model-level interpretation is defined with two aspects. First, model-level interpretation aims to figure out what has been learned by intermediate components in a trained model [65; 35], or what is the meaning of different locations in latent space [66; 67; 68]. Second, given an input instance, model-level interpretation unveils how the input is encoded by those components as latent representation [66; 67; 23; 25]. In our discussion, the former does not rely on input instances, while the latter is the opposite. Therefore, we name the two aspects as *Model Component Interpretation* and *Representation Interpretation* respectively to further distinguish them.

## 4.1 Model Component Interpretation for Understanding Adversarial Attacks

In deep models, model component interpretation can be defined as exploring the visual or semantic meaning of each neuron. A popular strategy to solve this problem is to recover the patterns that activate the neuron of interests at a specific layer [69; 35]. Following the previous notations, let $h(\mathbf{x})$ denote the activation degree of neuron $h$ given input $\mathbf{x}$, the perceived pattern of the neuron can be visualized by

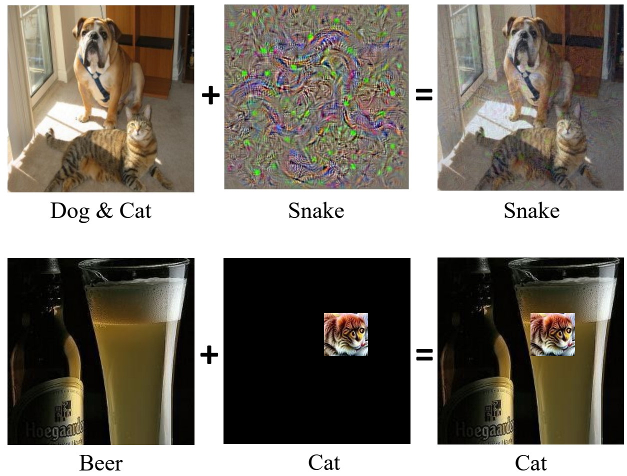


Figure 6: Examples of adversarial attacks after applying model-level interpretation. Upper: Targeted universal perturbation. Lower: Universal replacement attack.

solving the problem below:

$$\underset{\mathbf{x}'}{\operatorname{argmax}}\, h(\mathbf{x}') - \alpha \cdot C(\mathbf{x}'), \tag{17}$$

where $C(\cdot)$ such as $\|\cdot\|_1$ or $\|\cdot\|_2$ acts as regularization. Conceptually, the result contains patterns that neuron $h$ is sensitive to. If we choose $h$ to be $f_c$, then the resultant $\mathbf{x}'$ illustrates class appearances learned by the target model. Another discussion about different choices of $h$, such as neurons, channels, layers, logits and class probabilities, is provided in [70]. Similarly, we could also formulate another minimization problem

$$\underset{\mathbf{x}'}{\operatorname{argmin}}\, h(\mathbf{x}') + \alpha \cdot C(\mathbf{x}'), \tag{18}$$

to produce patterns that prohibit activation of certain model components or prediction towards certain classes.

The interpretation result $\mathbf{x}'$ is highly related with several types of adversarial attacks, with some examples shown in Figure 6.

- **Targeted-Universal-Perturbation Attack:** If we set $h$ to be class relevant mapping such as $f_c$, and solve Eq. 17 to get the interpretation, then $\mathbf{x}'$ can be directly added to target input instance as targeted perturbation attack. That is, given a clean input $\mathbf{x}_0$, the adversarial sample $\mathbf{x}^*$ is crafted simply as $\mathbf{x}^* = \mathbf{x}_0 + \lambda \cdot \mathbf{x}'$ to make $f(\mathbf{x}^*) = c$. It belongs to universal attack, because the interpretation process in Eq.17 does not utilize any information of the clean input. An example is shown in the upper row of Figure 6. A clean image is classified as "dog" (or "cat") by the model. Meanwhile, an image is generated by solving Eq.17, by setting $h$ as $f_c$ where $c$ denotes "snake". By adding the generated image to the clean image, the resultant image is recognized as "snake", although it still looks like a dog and a cat in human eyes.

- **Untargeted-Universal-Perturbation Attack:** If we set $h$ to be the aggregation of a number of middle-level layer mappings, such as $h(\mathbf{x}') = \sum_l \log(h^l(\mathbf{x}'))$ where $h^l$ denotes the feature map tensor at layer $l$, the resultant $\mathbf{x}'$ is expected to produce spurious activation to confuse the prediction of CNN models given any input, which implies $f(\mathbf{x}_0+\lambda\cdot\mathbf{x}') \neq f(\mathbf{x}_0)$ with high probability [13]. This can be seen as an untargeted and universal attack.

- **Universal-Replacement Attack:** Adversarial patches, which completely replace part of the input, represent a visually different attack from perturbation attack. Based on Eq.17, more parameters such as masks, shape, location and rotation could be considered in the optimization to control $\mathbf{x}'$ [71]. The patch is obtained as $\mathbf{x}' \odot \mathbf{m}$, and the adversarial sample $\mathbf{x}^* = \mathbf{x}_0 \odot (\mathbf{1}-\mathbf{m}) + \mathbf{x}' \odot \mathbf{m}$, where $\mathbf{m}$ is a binary mask that defines patch shape. Besides, recent research shows that, if we define $h$ as the objective score function in person detectors [14] or as the logit corresponding to human class [72], by solving Eq.18, it is possible to produce real-world patches attachable to human bodies to avoid them being detected by surveillance camera. An example of adversarial patches is shown in the bottom row of Figure 6. A clean image is classified as "beer". Meanwhile, a small image patch is generated, which is recognized as a "cat". By attaching the generated patch to the clean image, the prediction on the new image will be affected by the patch.

## 4.2 Representation Interpretation for Initiating Adversarial Attacks

Representation learning plays a crucial role in recent advances of machine learning, with applications in vision [73], natural language processing [74] and network analysis [75]. However, the opacity of representation space also becomes the bottleneck for understanding complex models. A commonly used strategy toward understanding representation is to define a set of explainable bases, and then decompose representation points according to the bases. Formally, let $\mathbf{z}_i \in \mathbb{R}^D$ denote a representation vector, and $\{\mathbf{b}_k \in \mathbb{R}^D\}_{k=1}^K$ denote the basis set, where $D$ denotes the representation dimension and $K$ is the number of base vectors. Then, through decomposition

$$\mathbf{z}_i = \sum_{k=1}^{K} p_{i,k} \cdot \mathbf{b}_k, \tag{19}$$

we can explain the meaning of $\mathbf{z}_i$ through referencing base vectors whose semantics are known, where $p_{i,k}$ measures the affiliation degree between instance $\mathbf{z}_i$ and $\mathbf{b}_k$. The work of providing representation interpretation following this scheme can be divided into several groups:

- **Dimension-wise Interpretation**: A straightforward way to achieve interpretability is to require each dimension to have a concrete meaning [76; 77], so that the basis can be seen as non-overlapping one-hot vectors. A natural extension to it would be to allow several dimensions (i.e., a segment) to jointly encode one meaning [78; 79].
- **Concept-wise Interpretation**: A set of high-level and intuitive concepts could first be defined, so that each $\mathbf{b}_k$ encodes one concept. Some examples include visual concepts [67; 66; 80], antonym words [81], and network communities [68].
- **Example-wise Interpretation**: Each base vector can be designed to match one data instance [82; 83] or part of the instance [84]. Those instances are also called prototypes. For example, a prototype could be an image region [84] or a node in networks [83].

The extra knowledge obtained from representation interpretation could be used to guide the direction of adversarial perturbation. However, the motivation of this type of work usually is to initiate more meaningful adversaries and then use adversarial training to improve model generalization, but not for the pure purpose of undermining model performance. For example, in text mining, [50] restricts perturbation direction of each word embedding to be a linear combination of vocabulary word embeddings, which improves model performance in text classification after adversarial training. In network embedding, [85] restricts perturbation of a node's embedding towards the embeddings of the node's neighbors in the network, so that adversarial training improves node classification and link prediction performances.

## 4.3 Model-Level Interpretation for Adversarial Defenses

Model-level interpretation develops an internal understanding of a model, including its weakness. Defenders could either choose to improve model robustness or develop a detector using internal data representation.

### 4.3.1 *Model Robustification With Model-Level Interpretation*

Some high-level features learned by deep models are not robust, which are insufficient to train reliable models. A novel algorithm is proposed in [86] to build datasets of robust features. Given a robust model $f_r$, $\mathcal{H}_r$ denotes the set of activations of neurons in the robust model, and $h : X \rightarrow \mathbb{R}$, $h \in \mathcal{H}_r$ is a transformation function that maps input to a neuron activation. Each instance in the robust dataset $\mathcal{D}_r$ is constructed from the original dataset $\mathcal{D}$. The instances in the robust dataset are expected to satisfy:

$$\mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}_r}[h(\mathbf{x})\cdot y] = \begin{cases} \mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}}[h(\mathbf{x})\cdot y], & \text{if } h \in \mathcal{H}_r \\ 0, & \text{otherwise} \end{cases}. \tag{20}$$

In this way, input information that corresponds to non-robust representations is suppressed. Instances in the robust dataset are expected to contain only the features that are relevant to the robust model.

Despite not being directly incorporated in model training, inspections of model-level interpretation, especially on latent representation, have motivated several defense methods. Through visualizing feature maps of latent representation layers, the noise led by adversarial perturbation can be easily observed [25; 23; 58]. With this observation, [25] proposes adding denoising blocks between intermediate layers of deep models, where the core function of the denoising blocks are chosen as low-pass filters. [23] observes that adversarial perturbation is magnified through feedforward propagation in deep models, and proposed a U-net model structure as denoiser. Furthermore, through neuron pattern visualization, [87] finds that the convolutional kernels of CNNs after adversarial training tend to show a more smooth pattern. Based on this observation, they propose to average each kernel weight with its neighbors in a CNN model, in order to improve the adversarial robustness.

### 4.3.2 *Adversarial Detection With Model-Level Interpretation*

Instead of training another large model as a detector using raw data, we can also leverage model-level interpretation to detect adversarial instances more efficiently. In this

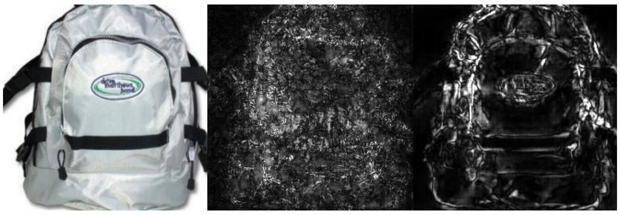

Figure 7: Explanations obtained from adversarially trained models focus less on textures and more on shape information [62]. Left: Input image. Middle: Gradient based explanation of a model without adversarial training. Right: Gradient based explanation of a model after adversarial training.

case, model-level interpretation plays a similar role as feature engineering, which helps distinguish between normal and adversarial instances. By regarding neurons as high-level features, readily available interpretation methods such as SHAP [88] could be applied for feature engineering to build adversarial detector [56]. After inspecting the role of neurons in prediction, a number of critical neurons could be selected. A steered model could be obtained by strengthening those critical neurons, while adversarial instances are detected if they are predicted very differently by the original model and steered model [29]. Nevertheless, the majority of work on adversarial detection utilizes latent representation of instances without inspecting their meanings, such as directly applying statistical methods on representations to build detectors [21; 89] or conducting additional coding steps on activations of neurons [28].

# 5. ADDITIONAL RELATIONS BETWEEN ADVERSARY AND INTERPRETATION

In the previous context, we have discussed how interpretation could be leveraged in adversarial attack and defense. In this section, we complement this viewpoint by analyzing the role of adversarial aspects of models in defining and evaluating interpretation. In addition, we specify the distinction between the two domains.

## 5.1 Improving Interpretation via Building Robust Models

In previous content, we have discussed the role of interpretation in studying model robustness. From another perspective, it has been found that, improving model robustness could also improve the understandability of explanations. First, the representations learned by robust models tend to align better with salient data characteristics and human perception [90]. Therefore, adversarially robust image classifiers are also useful in more sophisticated tasks such as generation, super-resolution, and translation [91], even without relying on GAN frameworks. Also, when attacking a robust classifier, the resultant adversarial samples are more likely to be recognized by humans [90]. In addition, retraining with adversarial samples [62], or regularizing gradients to improve model robustness [92], has been discovered to reduce noises from gradient-based sensitivity maps, and encourage CNN models to focus more on object shapes in making predictions. An example is shown in Figure 7. Finally, [93] presents the principle of "feature purification". The work discovers that dense mixtures of patterns exist in the weights of models trained with clean data using normal gradient descent. The dense pattern mixtures still generalize well when being used to predict normal data, but they are extremely sensitive to small perturbations in the input. After adversarial training, dense pattern mixtures could be removed, and the activation patterns of neurons will be easier to understand.

## 5.2 Defining Interpretation Approaches via Adversarial Perturbation

Some definitions of interpretation are inspired by adversarial perturbation. For feature-level interpretation, to understand the importance of a certain feature $x$, we try to answer a hypothetical question that "What would happen to the prediction Y, if $x$ is removed or distorted?". This is closely related to *causal inference* [94; 95], and samples crafted in this way are also called *counterfactual explanations* [96]. For example, to understand how different words in sentences contribute to downstream NLP tasks, we can erase the target words from input, so that the variation in output indicates whether the erased information is important for prediction [97]. In image processing, salient regions could be defined as the input parts that most affect the output value when perturbed [57]. Considering that using traditional iterative algorithms to generate masks is time-consuming, Goyal et al. [98] develops trainable masking models that generate masks in real time.

Besides defining feature-level interpretation, the similar strategy can be used to define model component interpretation. Essentially we need to answer the question that "How the model output will change if we change the component in the model?". The general idea is to treat the structure of a deep model as a causal model [99], or extract human understandable concepts to build a causal model [100], and then estimate the effect of each component via causal reasoning. The importance of a component is measured by computing output changes after the component is removed.

As a natural extension from the discussion above, adversarial perturbation can also be used to evaluate the interpretation result. For example, after obtaining the important features, and understanding whether they are positively or negatively related to the output, we could remove or distort these features to observe the target model's performance change [5; 45]. If the target model's performance significantly drops, then we are likely to have the correct interpretation. However, it is worth noting that the evaluation will not be fair if the metric and interpretation methods do not match [101].

## 5.3 Uniqueness of Model Explainability

Despite the common techniques applied for acquiring interpretation and exploring adversary characteristics, some aspects of the two directions put radically different requirements. For example, some applications require interpretation to be easily understood by human especially by AI novices, such as providing more user-friendly interfaces to visualize and present interpretation [102; 103; 104], while adversarial attack requires perturbation to be imperceptible to human. Some work tries to adapt interpretation to fit human cognition habits, such as providing example-based interpretation [105], criticism mechanism [106] and counterfactual explanation [107]. The emphasis of understandability in interpretability, by its nature, is exactly opposite to the main objective in adversarial attack, which is to add perturbation that is too subtle to be perceived by human.

## 6. CHALLENGES AND FUTURE WORK

We briefly introduce the challenges in leveraging interpretation to analyze the adversarial robustness of models. Meanwhile, we discuss the future research directions.

### 6.1 Models With Better Interpretability

Although interpretation could provide important directions against adversaries, interpretation techniques with better stability and faithfulness are needed before it could really be widely used as a reliable tool. As one of the challenges, it has been shown that many existing interpretation methods are vulnerable to manipulations [63; 64; 108]. A stable interpretation method, given an input instance and a target model, should produce relatively consistent results under the situation that the input may be subject to certain noises. As a preliminary work, [109] analyzes the phenomenon from a geometric perspective of decision boundary and proposed a smoothed activation function to replace ReLU. [110] proposes a sparsified variant of SmoothGrad [38] to produce saliency maps that is certifiably robust to adversaries.

Besides post-hoc interpretation, another challenge we are facing is how to develop models that are intrinsically interpretable [36]. With intrinsic interpretability, it may be more straightforward to identify and modify the undesirable aspects of model. Some preliminary work starts to explore applying graph-based models, such as proposing relational inductive biases to facilitate learning about entities and their relations [111], towards a foundation of an interpretable and flexible scheme of reasoning. Novel neural architectures have also been proposed, such as capsule networks [112] and causal models [113].

### 6.2 Adversarial Attacks in Real Scenarios

The most common scenario in existing work considers adversarial noises or patches in image classification or object detection. However, these types of perturbation may not represent the actual threats in the physical world. To solve the challenge, more realistic adversarial scenarios need to be studied in different applications. Some preliminary work include verification code generation[2], semantically or syntactically equivalent adversarial text generation [4; 114], and adversarial attack on graph data [6; 115]. Meanwhile, model developers need to be alerted to new types of attacks that utilize interpretation as the back door. For example, it is possible to build models that predict correctly on normal data, but make mistakes on input with certain secret attacker-chosen property [116]. Also, recently researchers found that it is possible to break data privacy by reconstructing private data merely from gradients communicated between machines [117].

### 6.3 Model Improvement Using Adversaries

The value of adversarial samples goes beyond simply serving as prewarning of model vulnerability. It is possible that the vulnerability to adversarial samples reflects some deeper generalization issues of deep models [118; 119]. Some preliminary work has been conducted to understand the difference between a robust model and a non-robust one. For example, it has been shown that adversarially trained models possess better interpretability [62] and representations with higher quality [91]. [120] also tries to connect adversarial robustness with model credibility, where credibility measures the degree that a model's reasoning conforms with human common sense. Another challenging problem is how to properly use adversarial samples to benefit model performance, since many existing works report that training with adversarial samples will lead to performance degradation, especially on large data [3; 25]. Recently, [121] shows that, by separately considering the distributions of normal data and adversarial data with batch normalization, adversarial training can be used to improve model accuracy.

[2]https://github.com/littleredhat1997/captcha-adversarial-attack

## 7. CONCLUSION

In this paper, we review the recent work of adversarial attacks and defenses by combining them with the recent advances of interpretable machine learning. Specifically, we categorize interpretation techniques into feature-level interpretation and model-level interpretation. Within each category, we investigate how the interpretation could be used for initiating adversarial attacks or designing defense approaches. After that, we briefly discuss other relations between interpretation and adversarial perturbation/robustness. Finally, we discuss the current challenges of developing transparent and robust models, as well as some potential directions to further study and utilize adversarial samples.

## 8. REFERENCES


[1] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[2] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[3] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. 2017.

[4] Qi Lei, Lingfei Wu, Pin-Yu Chen, Alexandros G Dimakis, Inderjit S Dhillon, and Michael Witbrock. Discrete adversarial attacks and submodular optimization with applications to text classification. *Systems and Machine Learning (SysML)*, 2019.

[5] Ninghao Liu, Hongxia Yang, and Xia Hu. Adversarial detection with model interpretation. In *KDD*, 2018.

[6] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. In *KDD*, 2018.

[7] Jian Kang and Hanghang Tong. N2n: Network derivative mining. In *CIKM*, 2019.

[8] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, and Tadayoshi Kohno. Physical adversarial examples for object detectors. In *12th {USENIX} Workshop on Offensive Technologies ({WOOT} 18)*, 2018.

[9] Mary Frances Zeager, Aksheetha Sridhar, Nathan Fogal, Stephen Adams, Donald E Brown, and Peter A Beling. Adversarial learning in credit card fraud detection. In *2017 Systems and Information Engineering Design Symposium (SIEDS)*, 2017.

[10] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 2017.

[11] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

[12] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *CVPR*, 2017.

[13] Konda Reddy Mopuri, Aditya Ganeshan, and Venkatesh Babu Radhakrishnan. Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE transactions on pattern analysis and machine intelligence*.

[14] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *CVPR Workshops*, 2019.

[15] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[16] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[17] Tania Lombrozo. The structure and function of explanations. *Trends in cognitive sciences*, 2006.

[18] Frank C Keil. Explanation and understanding. *Annu. Rev. Psychol.*, pages 227–254, 2006.

[19] Carl G Hempel and Paul Oppenheim. Studies in the logic of explanation. *Philosophy of science*, 1948.

[20] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 2018.

[21] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. *ICLR*, 2017.

[22] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017.

[23] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *CVPR*, 2018.

[24] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

[25] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *CVPR*, 2019.

[26] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[27] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE.

[28] Jiajun Lu, Theerasit Issaranon, and David Forsyth. Safetynet: Detecting and rejecting adversarial examples robustly. In *ICCV*, 2017.

[29] Guanhong Tao, Shiqing Ma, Yingqi Liu, and Xiangyu Zhang. Attacks meet interpretability: Attribute-steered detection of adversarial samples. In *NIPS*, 2018.

[30] Zhitao Gong, Wenlu Wang, and Wei-Shinn Ku. Adversarial and clean data are not twins. *arXiv preprint arXiv:1704.04960*, 2017.

[31] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017.

[32] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017.

[33] Mengnan Du, Ninghao Liu, Fan Yang, Shuiwang Ji, and Xia Hu. On attribution of recurrent neural network predictions via additive decomposition. In *The World Wide Web Conference*, 2019.

[34] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[35] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[36] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 2019.

[37] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Interpretable machine learning: definitions, methods, and applications. *arXiv preprint arXiv:1901.04592*, 2019.

[38] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[39] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[40] Xiaoyu Cao and Neil Zhenqiang Gong. Mitigating evasion attacks to deep neural networks via region-based classification. In *ACSAC*, 2017.

[41] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *ICML*, 2017.

[42] Zhengping Che, Sanjay Purushotham, Robinder Khemani, and Yan Liu. Distilling knowledge from deep networks with applications to healthcare domain. *arXiv preprint arXiv:1512.03542*, 2015.

[43] Jun Gao, Ninghao Liu, Mark Lawley, and Xia Hu. An interpretable classification framework for information extraction from online healthcare forums. *Journal of healthcare engineering*, 2017.

[44] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *KDD*, 2016.

[45] Wenbo Guo, Dongliang Mu, Jun Xu, Purui Su, Gang Wang, and Xinyu Xing. Lemna: Explaining deep learning based security applications. In *CCS*, 2018.

[46] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, 2013.

[47] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. 2018.

[48] Adith Boloor, Xin He, Christopher Gill, Yevgeniy Vorobeychik, and Xuan Zhang. Simple physical adversarial examples against end-to-end autonomous driving models. In *ICESS*. IEEE, 2019.

[49] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, and Payel Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *NIPS*, 2018.

[50] Motoki Sato, Jun Suzuki, Hiroyuki Shindo, and Yuji Matsumoto. Interpretable adversarial perturbation in input embedding space for text. In *IJCAI*, 2018.

[51] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *ICML*, 2017.

[52] Yaojing Wang, Yuan Yao, Hanghang Tong, Feng Xu, and Jian Lu. Discerning edge influence for network embedding. In *CIKM*, 2019.

[53] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.

[54] Nicholas Baker, Hongjing Lu, Gennady Erlikhman, and Philip J Kellman. Deep convolutional networks do not classify based on global object shape. *PLoS computational biology*, 2018.

[55] Baifeng Shi, Dinghuai Zhang, Qi Dai, Zhanxing Zhu, Yadong Mu, and Jingdong Wang. Informative dropout for robust representation learning: A shape-bias perspective. 2020.

[56] Gil Fidel, Ron Bitton, and Asaf Shabtai. When explainability meets adversarial learning: Detecting adversarial examples using shap signatures. *arXiv preprint arXiv:1909.03418*, 2019.

[57] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *ICCV*, 2017.

[58] Puyudi Yang, Jianbo Chen, Cho-Jui Hsieh, Jane-Ling Wang, and Michael I Jordan. Ml-loo: Detecting adversarial examples with feature attribution. *arXiv preprint arXiv:1906.03499*, 2019.

[59] Chiliang Zhang, Zuochang Ye, Yan Wang, and Zhimou Yang. Detecting adversarial perturbations with saliency. In *2018 IEEE 3rd International Conference on Signal and Image Processing (ICSIP)*, pages 271–275. IEEE, 2018.

[60] Jingyuan Wang, Yufan Wu, Mingxuan Li, Xin Lin, Junjie Wu, and Chao Li. Interpretability is a kind of safety: An interpreter-based ensemble for adversary defense. In *KDD*, 2020.

[61] Yinpeng Dong, Hang Su, Jun Zhu, and Fan Bao. Towards interpretable deep neural networks by leveraging adversarial examples. *arXiv preprint arXiv:1708.05493*, 2017.

[62] Tianyuan Zhang and Zhanxing Zhu. Interpreting adversarially trained convolutional neural networks. In *International Conference on Machine Learning*, pages 7502–7511, 2019.

[63] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *AAAI*, 2019.

[64] Akshayvarun Subramanya, Vipin Pillai, and Hamed Pirsiavash. Fooling network interpretation in image classification. In *ICCV*, 2019.

[65] Quanshi Zhang, Yu Yang, Haotian Ma, and Ying Nian Wu. Interpreting cnns via decision trees. In *CVPR*, 2019.

[66] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *ICML*, 2018.

[67] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Interpretable basis decomposition for visual explanation. In *ECCV*, 2018.

[68] Ninghao Liu, Xiao Huang, Jundong Li, and Xia Hu. On interpretation of network embedding via taxonomy induction. In *KDD*, 2018.

[69] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, page 1.

[70] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

[71] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.

[72] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *CCS*, 2016.

[73] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 2013.

[74] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *IEEE Computational intelligenCe magazine*, 2018.

[75] William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.

[76] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2017.

[77] Abhishek Panigrahi, Harsha Vardhan Simhadri, and Chiranjib Bhattacharyya. Word2sense: Sparse interpretable word embeddings. In *ACL*, 2019.

[78] Ninghao Liu, Qiaoyu Tan, Yuening Li, Hongxia Yang, Jingren Zhou, and Xia Hu. Is a single vector enough? exploring node polysemy for network embedding. In *KDD*, 2019.

[79] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. Learning disentangled representations for recommendation. In *Advances in Neural Information Processing Systems*, 2019.

[80] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. In *NeurIPS*, 2019.

[81] Binny Mathew, Sandipan Sikdar, Florian Lemmerich, and Markus Strohmaier. The polar framework: Polar opposites enable interpretability of pre-trained word embeddings. In *The World Wide Web Conference*, 2020.

[82] Been Kim, Cynthia Rudin, and Julie A Shah. The bayesian case model: A generative approach for case-based reasoning and prototype classification. In *NIPS*, 2014.

[83] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[84] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. In *NeurIPS*, 2019.

[85] Quanyu Dai, Xiao Shen, Liang Zhang, Qiang Li, and Dan Wang. Adversarial training methods for network embedding. In *The World Wide Web Conference*, 2019.

[86] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019.

[87] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *CVPR*, 2020.

[88] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *NIPS*, 2017.

[89] Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

[90] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

[91] Shibani Santurkar, Andrew Ilyas, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Image synthesis with a single (robust) classifier. In *NeurIPS*, 2019.

[92] Ninghao Liu, Xiao Huang, Jundong Li, and Xia Hu. On interpretation of network embedding via taxonomy induction. In *KDD*, 2018.

[93] Zeyuan Allen-Zhu and Yuanzhi Li. Feature purification: How adversarial training performs robust deep learning. *arXiv preprint arXiv:2005.10190*, 2020.

[94] Ruocheng Guo, Lu Cheng, Jundong Li, P Richard Hahn, and Huan Liu. A survey of learning causality with data: Problems and methods. *ACM Computing Surveys (CSUR)*, 2020.

[95] Raha Moraffah, Mansooreh Karami, Ruocheng Guo, Adrienne Raglin, and Huan Liu. Causal interpretability for machine learning-problems, methods and evaluation. *ACM SIGKDD Explorations Newsletter*, 2020.

[96] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[97] Jiwei Li, Will Monroe, and Dan Jurafsky. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*, 2016.

[98] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. In *NIPS*, 2017.

[99] Tanmayee Narendra, Anush Sankaran, Deepak Vijaykeerthy, and Senthil Mani. Explaining deep learning models using causal inference. *arXiv preprint arXiv:1811.04376*, 2018.

[100] Michael Harradon, Jeff Druce, and Brian Ruttenberg. Causal learning and explanation of deep neural networks via autoencoded activations. *arXiv preprint arXiv:1802.00541*, 2018.

[101] Ninghao Liu, Yunsong Meng, Xia Hu, Tie Wang, and Bo Long. Are interpretations fairly evaluated? a definition driven pipeline for post-hoc interpretability. *arXiv preprint arXiv:2009.07494*, 2020.

[102] Brian Y Lim, Anind K Dey, and Daniel Avrahami. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2009.

[103] Menaka Narayanan, Emily Chen, Jeffrey He, Been Kim, Sam Gershman, and Finale Doshi-Velez. How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation. *arXiv preprint arXiv:1802.00682*, 2018.

[104] Fan Yang, Shiva K Pentyala, Sina Mohseni, Mengnan Du, Hao Yuan, Rhema Linder, Eric D Ragan, Shuiwang Ji, and Xia Ben Hu. Xfake: Explainable fake news detector with visualizations. In *WWW*, 2019.

[105] Isabelle Bichindaritz and Cindy Marling. Case-based reasoning in the health sciences: What's next? *Artificial intelligence in medicine*, 2006.

[106] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *NIPS*, 2016.

[107] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *ICML*, 2019.

[108] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *AAAI*, 2020.

[109] Ann-Kathrin Dombrowski, Maximillian Alber, Christopher Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. Explanations can be manipulated and geometry is to blame. In *NeurIPS*, 2019.

[110] Alexander Levine, Sahil Singla, and Soheil Feizi. Certifiably robust interpretation in deep learning. *arXiv preprint arXiv:1905.12105*, 2019.

[111] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[112] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *NIPS*, 2017.

[113] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. MIT press, 2017.

[114] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, 2018.

[115] Qinghai Zhou, Liangyue Li, Nan Cao, Lei Ying, and Hanghang Tong. Admiring: Adversarial multi-network mining. In *ICDM*, 2019.

[116] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

[117] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *arXiv preprint arXiv:1906.08935*, 2019.

[118] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *NeurIPS*, 2019.

[119] Tom B Brown, Nicholas Carlini, Chiyuan Zhang, Catherine Olsson, Paul Christiano, and Ian Goodfellow. Unrestricted adversarial examples. *arXiv preprint arXiv:1809.08352*, 2018.

[120] Mengnan Du, Ninghao Liu, Fan Yang, and Xia Hu. Learning credible deep neural networks with rationale regularization. In *ICDM*, 2019.

[121] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan Yuille, and Quoc V Le. Adversarial examples improve image recognition. *arXiv preprint arXiv:1911.09665*, 2019.