2022

Reverse Engineering of Adversarial Samples by Leveraging Patterns left by the Attacker

Rahul Ambati
University of Central Florida

REVERSE ENGINEERING OF ADVERSARIAL SAMPLES BY LEVERAGING PATTERNS LEFT BY THE ATTACKER

by

RAHUL AMBATI
M.S. International Institute of Information Technology, 2019

A thesis submitted in partial fulfilment of the requirements
for the degree of Master of Science
in the Department of Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2022

Major Professor: Yogesh Singh Rawat

ABSTRACT

Intrinsic susceptibility of deep learning to adversarial examples has led to a plethora of attack techniques with a common broad objective of fooling deep models. However, we find slight compositional differences between the algorithms achieving this objective. These differences leave traces that provide important clues for attacker profiling in real-life scenarios. Inspired by this, we introduce a novel problem of *'Reverse Engineering of aDversarial attacks' (RED)*. Given an adversarial example, the objective of RED is to identify the attack used to generate it. Under this perspective, we can systematically group existing attacks into different families, leading to the sub-problem of attack family identification. To enable RED analysis, we introduce a large *'Adversarial Identification Dataset' (AID)*, comprising over 180k adversarial samples generated with 13 popular attacks for image specific/agnostic white/black box setups. We use AID to devise a novel framework for the RED objective. The proposed framework is designed using a novel Transformer based Global-LOcal Feature(GLoF) module which helps in approximating the adversarial perturbation and identification of the attack. Using AID and our framework, we provide multiple interesting benchmark results for the RED problem.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

Deep learning is currently at the center of many emerging technologies, from autonomous vehicles to smart surveillance and numerous security applications. However, it is now also well-established that deep neural networks are susceptible to adversarial attacks [1, 9]. This intriguing weakness of deep learning, which is otherwise known to supersede human intelligence in complex tasks [50], has attracted an ever-increasing interest of the research community in the last few years [10, 2]. This has led to a wide range of adversarial attacks that can effectively fool deep learning models. Although adversarial attacks have also led to research in defenses, there is a consensus that defenses currently lack efficacy [2]. Many of them are easily broken, or become ineffective by changing the attack strategy [2].

Incidental, deep learning in practice is still widely open to malicious manipulation through adversarial attacks [1, 9]. It is yet to be seen if this technology can retain its impressive performance while also demonstrating robustness to adversarial attacks. Until an intrinsically (adversarially) robust high-performing deep learning framework is developed, practitioners must account for the adversarial susceptibility of deep learning in all applications. These conditions give rise to an important practical problem of 'attacker profiling'. In real-life, understanding the attacker's abilities can allow counter-measures even outside the realm of deep learning. However, the current literature on adversarial attacks on deep learning is almost completely void of any exploration along this line [2]. From the pragmatic viewpoint, the primal question of this potential research direction is, *"given an adversarial example, which attack algorithm was used to generate it?"*.

In this work, we take the first systematic step towards answering this question with Reverse Engineering of aDversaial attacks (RED). Focusing on the *additive adversarial perturbations*, our aim is to explore the extent to which a victim is able to identify its attacker by analysing only the

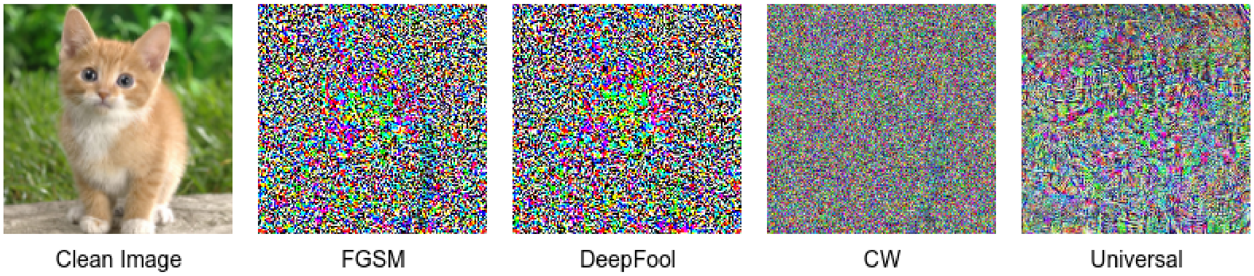


Figure 1.1: Despite their imperceptibility, adversarial perturbations contain peculiar patterns.

adversarial input. To explore this new direction, it is imperative to curate a large database of adversarial samples. To that end, we introduce Adversarial Identification Dataset (AID) which consists of over 180k adversarial samples, generated with 13 popular attacks in the literature. The dataset covers input-specific and input-agnostic attacks, and considers white- and black-box setups. We select these attacks considering the objective of our reverse engineering problem.

We use AID to explore RED with a proposed framework that is built on the intuition that attack algorithms leave their peculiar signatures in the adversarial examples. As seen in Fig. 1.1, these traces can reveal interesting information that can help in profiling the attacker. Our technique works on the principle of extracting those signatures. At the center of our framework is a signature extractor which is trained to extract input-specific signatures. Unlike random noise, these traces contain global as well as local structure. Motivated, we design the signature extractor consisting GLoF modules that combine CNN's ability to learn local structure [31] and transformer's capability to capture global information [55, 56, 18]. These signatures contain information relating to the attack algorithm. We classify the extracted signature to identify the attack leveraged to generate the adversarial example. We benchmark the effectiveness of the proposed framework on AID.

Our contributions are summarized as follow.

- We put forth a new problem of Reverse Engineering of aDversarial attacks (RED), which is aimed at profiling the attacker generating the adversarial input. We formalize RED to provide a systematic guideline for research in this direction.

- We propose an effective framework to provide the first-of-its-kind study for the RED problem.
- We introduce a large Adversarial Identification Dataset (AID), comprising 180k+ adversarial samples generated with 13 attacks. AID is used to extensively study RED, leading to promising results.

CHAPTER 2: RELATED WORK

Adversarial attacks and defenses is currently a highly active research direction. For a comprehensive review of the recent literature in this direction, we refer to [2]. Our discussion here focuses on the relevant aspects of this direction with representative existing techniques.

The discovery of adversarial susceptibility of deep learning was made in the context of visual classifiers [52]. Szegedy et al. [52] demonstrated that deep models can be fooled into incorrect prediction by adding imperceptible adversarial perturbations to the input. They also showed that by adding such adversarial samples in the training data, models can achieve robustness to adversarial manipulation. Hence, to efficiently compute adversarial samples (for adversarial training), [20] proposed the Fast Gradient Sign Method (FGSM). Conceptually, the FGSM takes a single gradient ascend step over the loss surface of the model w.r.t. input to compute the adversarial perturbation. This notion of gradient ascend is one of the most popular strategy to compute adversarial examples in the relevant literature.

Kurakin et al. [30] enhanced FGSM to iteratively take multiple small steps for gradient ascend, thereby calling their strategy Basic Iterative Method (BIM). A similar underlying scheme is adopted by the Projected Gradient Descent (PGD) attack [37], with an additional step of projecting the gradient signals on a pre-fixed ℓ_p-ball to constrain the norm of the resulting perturbation signal. All the above attacks must compute model gradient to compute the perturbations. Hence, we can categorise them as gradient-based attacks. Moreover, the gradient computation normally requires complete knowledge of the model itself. In the parlance of adversarial machine learning, the broad category of these attacks is known as white-box attacks [1]. Other popular attacks that can be categorised as white-box attacks include Carlini & Wagner attack [7], DeepFool [41] and Jacobian Saliency Map Attack (JSMA) [43]. The counterpart of the white-box attack category is the black-box category. Black-box attacks do not assume any knowledge of the target model, except its predictions. The most popular stream of black-box attacks are query-based attacks, which allow the attacker to iteratively refine an adversarial example by sending the current version to the remote model as a query. The model's prediction is used as a feedback for improving the adversarial nature of the input. If the attacker only receives the model decision (not its confidence score), then such a query-based attack is called a decision-based attack. Currently, the decision based attacks are more popular in black-box setups due to their pragmetic nature. A few recent representative examples in this category include [46], [49], [19], [32].

With the ever increasing number of attack techniques, there is a considerable interest of the research community in devising defences against the attacks. To that end, adversarial training is one of the most popular strategies [20, 27, 37, 54, 61]. It trains the model on adversarial images themselves to make it more robust. Another line of research in adversarial defenses transforms the input image to reduce the adversarial effects of the embedded perturbations [21, 59, 57]. These transformations also include image denoising to make adversarial image benign [35, 58]. [27] proposed a variant of adversarial training that trains the network to minimize the the distance of its predictions for benign and their corresponding adversarial images. While the defense mechanisms are making significant efforts in suppressing the nature of perturbations, there have always been counter attacks that can fool these networks.

The existing literature also covers a wide range of other defense techniques, from augmenting the models with external defense modules [45, 33, 14] to certified defenses [28, 53, 13]. We refer interested reader to [2] for recent advances in this direction. Here, we emphasize that although effective, these defenses generally come at considerable computational cost and degradation in model performance on clean inputs. This makes them less appealing for real-world applications.

Instead of proposing yet another defense, we take a different perspective on addressing the adversarial susceptibility of deep learning. Assuming a deployed model, we aim at identifying the capabilities of the attacker. Such an attacker profiling can help in adversarial defenses outside the realm of deep learning. This is more practical because it can eventually allow deep learning models to disregard intrinsic/appended defensive modules that result in performance degradation, causing deep learning to lose its advantage over other machine learning frameworks.

CHAPTER 3: THE RED PROBLEM

The Reverse Engineering of aDversarial attacks (RED) problem is generic in nature. However, we limit its scope to visual classifiers in this work for a systematic first-of-its-kind study. Let $\mathcal{C}(.)$ be a deep visual classifier such that $\mathcal{C}(\mathbf{I}) : \mathbf{I} \rightarrow \boldsymbol{\ell}$, where $\mathbf{I} \in \mathbb{R}^m$ is a natural image and $\boldsymbol{\ell} \in \mathbb{Z}^+$ is the output of the classifier. For attacking $\mathcal{C}(.)$, an adversary seeks a signal $\boldsymbol{\rho} \in \mathbb{R}^m$ to achieve $\mathcal{C}(\mathbf{I} + \boldsymbol{\rho}) \rightarrow \tilde{\boldsymbol{\ell}}$, where $\tilde{\boldsymbol{\ell}} \neq \boldsymbol{\ell}$. To ensure that the manipulation to a clean image is humanly imperceptible, the perturbation $\boldsymbol{\rho}$ is norm-bounded, e.g., by enforcing $||\boldsymbol{\rho}||_p < \eta$, where $||.||_p$ denotes the ℓ_p-norm of a vector and 'η' is a pre-defined scalar. More concisely, the adversary seeks $\boldsymbol{\rho}$ that satisfies

$$\mathcal{C}(\mathbf{I} + \boldsymbol{\rho}) \rightarrow \tilde{\boldsymbol{\ell}} \text{ s.t. } \tilde{\boldsymbol{\ell}} \neq \boldsymbol{\ell}, ||\boldsymbol{\rho}||_p < \eta. \tag{3.1}$$

The above formulation underpins the most widely adopted settings for the adversarial attacks, where $\boldsymbol{\rho}$ is a systematically computed additive signal. From our RED perspective, we see this signal as a function $\boldsymbol{\rho}(\mathcal{A}, \{\mathbf{I}\}, \mathcal{C})$, where $\mathcal{A}$ identifies the algorithm used to generate the perturbation and $\{\mathbf{I}\}$ indicates that $\boldsymbol{\rho}$ can be defined over a set of images instead of a single image, e.g., in universal perturbations [39].

In practice, the targeted model $\mathcal{C}$ must already be deployed and the input $\mathbf{I}$ fixed during an attack. This leaves $\mathcal{A}$ as the main point of interest for the RED problem. For clarity, we often refer to $\mathcal{A}$ directly as 'attack' in the text. To abstract away the algorithmic details, we can conceptualize $\mathcal{A}$ as a function $\mathcal{A}(\{\boldsymbol{\varphi}\}, \{\boldsymbol{\psi}\})$, where $\{\boldsymbol{\varphi}\}$ denotes a set of abstract design hyper-parameters and $\{\boldsymbol{\psi}\}$ is a set of numeric hyper-parameters. To exemplify, the choice of the scope of the adversarial objective, e.g. universal vs image-specific, is governed by an element in $\{\boldsymbol{\varphi}\}$. Similarly, the choices of 'η' or 'p' values in Eq. (3.1) are overseen by the elements of $\{\boldsymbol{\psi}\}$. Collectively, both sets contain all the hyper-parameters available to an attacker to compute $\boldsymbol{\rho}$.

The numeric hyper-parameter set $\{\boldsymbol{\psi}\}$ is relatively less interesting because a simple choice of numeric value does not help in profiling the attacker, which is the ultimate objective of the RED problem. We are particularly interested in the design choices made under $\{\boldsymbol{\varphi}\}$. In the considered settings, $\{\boldsymbol{\varphi}\}$ is a finite set because each of its elements, i.e., $\varphi_i \in \{\boldsymbol{\varphi}\}$, governs a choice along a specific design dimension under the practical constraint that the attack must achieve its fooling objective. Nevertheless, in this work, we are not after exhaustively listing the elements of $\{\boldsymbol{\varphi}\}$. Instead, we specify only three representative elements to demonstrate the possibility of attack reverse engineering. These three elements are:

- φ_1 : Model gradient information.
- φ_2 : Black-box prediction score information.
- φ_3 : Attack fooling scope.

It is possible to easily extend the above list to incorporate further design choices. The criterion for a parameter to be enrolled in $\{\boldsymbol{\varphi}\}$ is that a single choice should cover a range of existing attacks. For instance, φ_1 can either be `true` or `false`. The choice `true` can result in a family of attacks $\mathcal{F}_1^a$ of gradient-based attacks, covering FGSM [20], PGD [37], BIM [30] etc. Non-gradient based attack family $\mathcal{F}_1^b$ results when $\varphi_1 = \texttt{false}$. Similarly, when $\varphi_2 = \texttt{true}$, we get an attack family $\mathcal{F}_2^a$ of score-based black-box attacks[36, 25], and $\varphi_2 = \texttt{false}$ yields $\mathcal{F}_2^b$ that represents decision-based attacks[5, 4, 12]. We let $\varphi_3 \sim \{\texttt{universal}, \texttt{input-specific}\}$.

In the above formalism, $\mathcal{F}_i^x \cap \mathcal{F}_i^y = \emptyset$ always holds for the resulting attack families. However, we must allow $\mathcal{F}_i^x \cap \mathcal{F}_j^x \neq \emptyset$ because an attack family resulting from φ_i may still make choices for $\varphi_{j \neq i}$ without any constraint. For instance, a universal attack can be either gradient-based [39] or non-gradient-based [16]. This also indicates that our set $\{\boldsymbol{\varphi}\}$ is decided by the pool of known attacks themselves. This set can not be forced to be an exhaustive list of design parameters, because by itself, a simple choice of parameter value does not represent a fully functional attack family.

Let $\mathcal{F}_i = \{f_1^i, f_2^i, ..., f_Z^i\}$ denote the i^{th} attack family with 'Z' adversarial attacks that are formed under φ_i such that all $f_z^i \in \mathcal{F}_i$ satisfy the constraint in Eq. (3.1). Then, $f_z^i(\mathbf{I}) \rightarrow \tilde{\mathbf{I}}$ s.t. $\mathcal{C}(\tilde{\mathbf{I}}) \rightarrow \tilde{\boldsymbol{\ell}} \neq \boldsymbol{\ell}, ||\boldsymbol{\rho}||_p < \eta$. In this setting, the core RED problem is a reverse mapping problem that computes $\Psi(\tilde{\mathbf{I}}) \rightarrow f_z^i$, given a set of '$N$' attack families $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_N\}$. We must seek $\Psi(.)$ to solve this.

CHAPTER 4: ADVERSARIAL IDENTIFICATION DATASET(AID)

To investigate the RED problem, we develop Adversarial Identification Dataset (AID). This dataset is leveraged to explore the attack reverse engineering technique developed in this work. Below, we detail different attacks $\mathcal{A}$, attack families $\mathcal{F}$ and their design and numeric hyper-parameters $(\{\varphi\}, \{\psi\})$ considered in AID.

Overview

Most of the existing literature in adversarial attacks concentrates on devising novel attack schemes or robustifying models against the attacks. Multiple existing adversarial attack libraries such as advertorch [17], Adversarial robustness toolbox[42] and foolbox [47], etc., are available to generate adversarial samples on-the-fly. However, for our problem, it is imperative that we store the generated adversarial perturbations to analyze them for reverse engineering. This motivates the curation of Adversarial Identification Dataset (AID) that comprises perturbations generated by leveraging different attack strategies over a set of images targeting different pre-trained classifiers.

We make use of the existing adversarial attack libraries to generate the desired perturbations. Given a large number of adversarial attack methods and families available, we consider several factors while choosing which attacks are to be included in AID. Firstly, our choice is to include the most popular and well-studied attack techniques in the existing literature. Secondly, we prefer to have diversity in the perturbations by considering attack toolchain families that differ in the way they have access to the model(this includes model parameters, gradients, weights, etc.). Thirdly, we chose to consider attacks that generalize over the dataset. The resulting dataset consists of:

- 180k perturbations

Table 4.1: AID statistics

Parameter	**Details**
Dataset size	187.2k
Training samples	156k
Testing samples	31.2k
Per network train set	52k
Per network test set	10.4k
Total attacks	13
Toolchain families	3
Target networks	3
$\|\|\boldsymbol{\rho}\|\|_\infty$ range	$\{1, 16\}$
$\|\|\boldsymbol{\rho}\|\|_2$ range	$\{1, 10\}$

- three different toolchain families
- 13 different adversarial attacks

Attack Families

Gradient based attacks: As per our definition, gradient based attacks are the attacks that are able to exploit the gradients of the target model to perturb input images. Since the attacker needs access to the gradients, these attacks are typically white box in nature. Our gradient-based attack family consists of *Fast Gradient Sign Method (FGSM)* [20], *Basic Iterative Method*[30], *NewtonFool*[26], *Projected Gradient Descent(PGD)*[37], *DeepFool*[38], *Carlini Wagner (CW)*[8] attacks.

Decision based attacks: For our exploration, decision-based attacks are the fooling techniques applied in black-box setups where the attacker only has access to the decision of the target model. The attacker repeatedly queries the target model and utilizes the decision of the model to curate the perturbation. We consider *Additive Gaussian Noise*[47], *Gaussian Blur*[47], *Salt & Pepper Noise*[47], *Contrast Reduction*[47], *Boundary Attack*[6] for the decision-based attack family.

Table 4.2: Summary of the attacks considered in AID.

Attack Method	**Family**	**Setup**	**NB**
PGD [37]	Grad.	WB	l_∞
BIM [30]	Grad.	WB	l_∞
FGSM [20]	Grad.	WB	l_∞
DeepFool [38]	Grad.	WB	l_∞
NewtonFool [26]	Grad.	WB	l_2
CW [8]	Grad.	WB	l_2
Additive Gaussian [47]	Grad.	BB	l_2
Gaussian Blur [47]	Grad.	BB	l_∞
Salt&Pepper [47]	Grad.	BB	l_∞
Contrast Reduction [47]	Dec.	BB	l_∞
Boundary [6]	Dec.	BB	l_2
UAN [22]	Uni.	WB	l_∞
UAP [40]	Uni.	WB	l_∞

Universal attacks: Universal attacks generalize across a dataset. A single perturbation is sufficient to fool the network across multiple images with a desired fooling probability. Most common approaches to generate universal perturbations either iteratively compute perturbations by gradually computing and projecting model gradients over input batches, or use generative modelling to compute image agnostic perturbations. We consider *Universal Adversarial Perturbation (UAP)*[40], *Universal Adversarial Network (UAN)*[22] for the universal attack family.

Dataset creation

Benign samples: We require clean images to create an adversarial perturbation. We utilize ImageNet2012 [48] validation set consisting of 50k images spanning across 1000 classes. We split the validation set into two parts, forming training and test partitions of AID. This ensures that there the two partitions are mutually exclusive. Training set of perturbed images for AID is generated by randomly choosing 4k images per network per attack from the training partition. Similarly, the test set of perturbed images is generated by randomly choosing 800 images per network per attack from the test partition. Note that each attack image can be computed with different networks i.e. target models. We discuss these in the following section.

Target models: We consider three different target models; ResNet50 [23], DenseNet121 [24] and InceptionV3 [51]. The use of multiple models ensures that the adversarial samples are not specific to a target model.

Attack settings: In practice, there can be variations in perturbations norm for an attack - a hyper-parameter from $\{\psi\}$. This variation is incorporated in AID by sampling η from a range of values. For attacks constructed under l_∞ norm, we consider a range of $\{1, 16\}$ and $\{1, 10\}$ for l_2 norm based attacks. In Table 4.1, we provide summary statistics of the dataset. The procedure of generating the full dataset is further explained in the supplementary material of the paper. We also summarise the considered attacks, their families and used perturbation norm-bounds in Table 4.2.

CHAPTER 5: PROPOSED APPROACH

In this section, we discuss the design choices we consider for solving the RED problem $\Psi(\tilde{\mathbf{I}}) \rightarrow f_z^i$. A straightforward approach could be to build a classifier $C(\tilde{\mathbf{I}}) \rightarrow f_z^i$ that identifies the attack leveraged to generate the adversarial input $\tilde{\mathbf{I}}$. In such a scenario, the underlying patterns in the perturbation aren't preserved since the perturbation ρ is closely intertwined with the benign sample $\mathbf{I}$, thus making the problem much harder. To solve this problem, we design a signature extractor $\Omega(\tilde{\mathbf{I}}) \rightarrow \tilde{\boldsymbol{\rho}}$ that generates a signature $\tilde{\boldsymbol{\rho}}$ from the adversarial input such that it lies close to the original perturbation $\boldsymbol{\rho}$ while preserving patterns helpful in identifying the attacker. The objective of the signature extractor is,

$$\Omega(\tilde{\mathbf{I}}) \rightarrow \tilde{\boldsymbol{\rho}},\ \ ||\tilde{\boldsymbol{\rho}} - \boldsymbol{\rho}||_2 = \boldsymbol{\delta},\ \ min(\boldsymbol{\delta}). \tag{5.1}$$

While the objective draws similarities with existing problems like denoising/deraining, signature extraction is relatively complex. Noise/rain pertaining to these tasks are localized in nature and are visually perceptible in most cases. Contrastively, adversarial perturbations are nearly imperceptible and contain global patterns that makes the problem extremely challenging and requires methods beyond standard techniques aimed at denoising and other low-level computer vision tasks.

The adversarial input also contains minute imprints left by the attacker. We utilize these imprints to complete extent by extracting features from the input image as well and fusing them with the generated signature. Fused signature is passed on to the attack classifier C that identifies the attack. The objective of the attack classifier is,

$$C(\tilde{\boldsymbol{\rho}}) \rightarrow f_z^i,\ \ where\ \ f_z^i(\mathbf{I}) \rightarrow \tilde{\mathbf{I}} \tag{5.2}$$

where, $\tilde{\boldsymbol{\rho}}$ is the generated signature, f_z^i is the z^{th} attack from the i^{th} toolchain family. Figure 5.1 shows an overview of the proposed approach highlighting the signature extractor and the attack classifier.

Signature Extractor

The Signature Extractor serves the purpose of extracting a signature that contains patterns specific to the attack. As shown in Fig.5.1, the signature extractor has two streams of information flow progressing through a series of GLoF blocks. Each stream is designed to capture local or global features along with feature sharing across them that helps in generating a rectified image. GloF block utilizes convolutional layers to extract local features while attention mechanism applied over image patches to help in attaining global connectivity. Conjunction of global and local features help reconstruct a rectified image that lies in the neighborhood of the clean image. Subtracting the rectified image from the input adversarial image yields the signature.

Consider an adversarial image $\tilde{I} \in \mathbb{R}^{H \times W \times 3}$ (*H*, *W* correspond to image height and width and 3 corresponds to the RGB channels). The standard GLoF block receives a series of token embeddings and a 2D feature map of the image. The input adversarial image $\tilde{I}$ is reshaped into a series of 2D patches $\tilde{I}_p \in \mathbb{R}^{N \times P^2 \times 3}$, where P is the height and width of each patch and N is the number of patches/tokens $N = HW/P^2$. The patches are flattened along the feature dimension $\tilde{I}_p \in \mathbb{R}^{N \times (P^2 \cdot 3)}$. The attention arm along the GLoF blocks expects a constant embedding dimension D_1, hence the patches are projected onto the embedding space of dimension D. As proposed in [], adding position embeddings $E \in \mathbb{R}^{N \times D_1}$ to the patch embeddings help in the retain the relative position of the patches in the 2D space. The resulting patch embedding is termed $T_0 \in \mathbb{R}^{N \times D_1}$ (0 referring to the intital feature level).

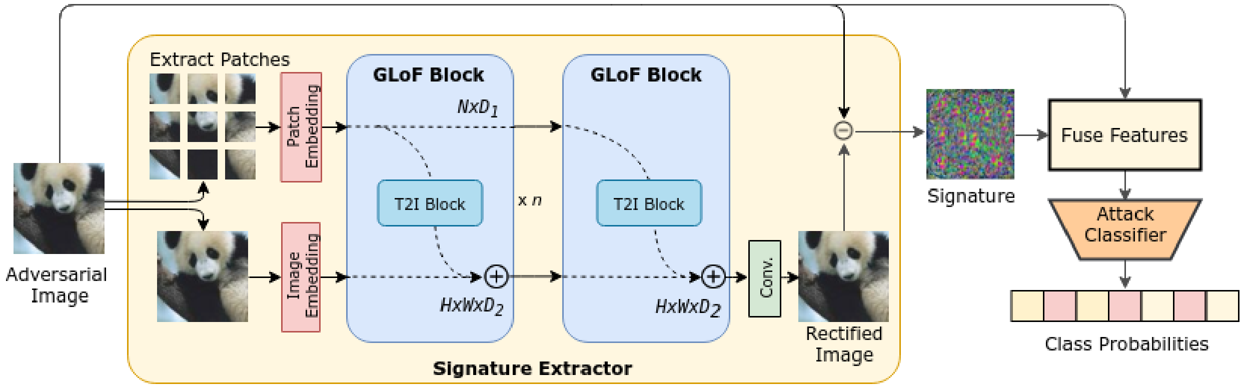


Figure 5.1: **Model Overview.**

$$T_0 = \tilde{I}_p + E; \quad \tilde{I}_p, E \in \mathbb{R}^{N \times D_1} \tag{5.3}$$

Alongside, the input image is projected to an embedding dimension $\boldsymbol{D_2}$, by applying a 3×3 Conv with D_2 features. We term these features $\boldsymbol{Z_0} \in \mathbb{R}^{H \times W \times D_2}$ (0 referring to the initial feature level). Features extracted from previous level$(l-1)$ are passed on to the next GLoF block.

$$\boldsymbol{T_l}, \boldsymbol{Z_l} = GLoF(\boldsymbol{T_{l-1}}, \boldsymbol{Z_{l-1}}); \quad l = 1...L \tag{5.4}$$

Where L is the number of GLoF blocks. The output of the final GLoF block corresponding to the convolutional arm $\boldsymbol{Z_l}$ is transformed to RGB space by applying a 3×3 Conv with 3 feature maps resulting in the rectified image $\boldsymbol{I_r} \in \mathbb{R}^{H \times W \times 3}$. Finally, to extract the signature from the rectified image, difference of the rectified and the original image is considered $\tilde{\boldsymbol{\rho}} = \tilde{\boldsymbol{I}} - \boldsymbol{I_r}$.

GLoF Module

Standard convolutional layers are very good at extracting local information [29]. However, a major drawback of convolutional layer is its inability to extract features from a receptive field larger than the kernel, hence missing global connectivity. On the other hand, transformers are known to be extremely powerful in learning non-local connectivity [15, 62]. As seen in [18], standard vision transformers splits the image into patches and learn the similarities between the patches. While this allows the network to learn global connectivity, it fails to utilize the local information [34, 60]. Overcoming these limitations, we propose Global-LOcal Feature extractor (GLoF) module to combine CNN's ability to extract low-level localized features and vision transformer's ability to extract global connectivity across long range tokens. Detailed schematic of the GLoF block is given in Fig.5.2. The GLoF block at any level receives the local and global features from the previous level. The GLoF block at level l receives the image features $Z_{l-1} \in \mathbb{R}^{H \times W \times D_2}$ and the embedded tokens $T_{l-1} \in \mathbb{R}^{N \times D_1}$ where $T_{l-1} = \{T_{l-1}^1, T_{l-1}^2, ..., T_{l-1}^N\}$ (N being the number of tokens) as inputs.

Local features: Embedded 2D image features from the previous layer $\boldsymbol{Z_{l-1}}$ are fed to a standard ResNet block with convolutional, batch norm and activation layers.

Global features: Embedded tokens are fed to attention mechanism. Series of tokens from previous layer $\boldsymbol{T_{l-1}}$ are passed through a multi-head attention layer which calculates the weighted sum. A feed forward network is applied over the attention output. It consists of two dense layers that are applied to individual tokens separately with GELU activation applied over the output of the first dense layer[18].

T2I Block: Features learned from the attention arm corresponding to the global connectivity are merged with the convolutional arm. **Token to Image (T2I)** block is responsible for rearranging the

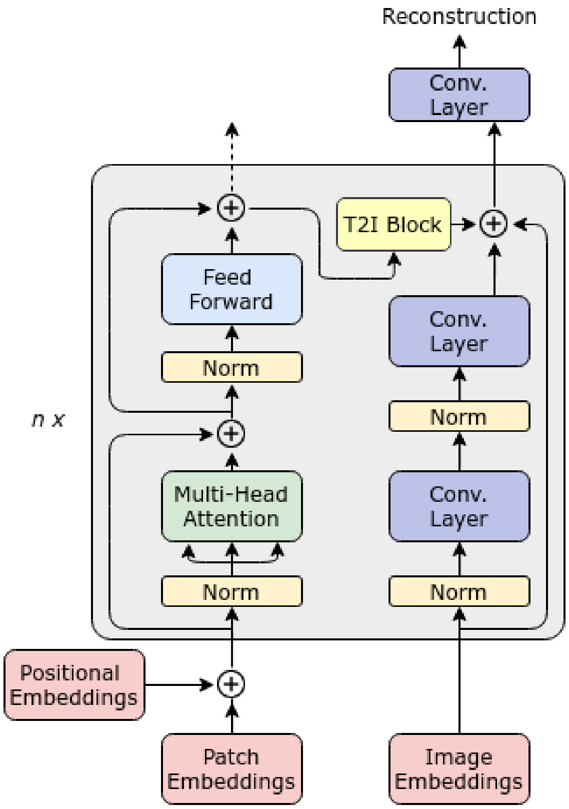


Figure 5.2: **GLoF block architecture.**

series of tokens to form a 2D grid. This transformed grid is passed on to a series of convolutional layers to obtain the feature map with the desired depth and is merged with the features from the convolution arm of the GLoF block. The merged features as well as the learned token embeddings are passed to consecutive GLoF blocks.

Attack Classifier

The generated signature is specific to the input image. Since the input image also contains imprints pertaining to the attacker, we complement the extracted signature with the adversarial input and feed it to the attack classifier. The fusion is done by applying a series of convolutional layers over the signature and the input image separately and concatenating them. We explore several standard pre-trained CNN networks as attack classifiers.

Training Objective

We use L_2 loss to minimize the the distance of the generated signature $\tilde{\boldsymbol{\rho}}$ to the raw perturbation $\boldsymbol{\rho}$. Alongside, the attack classifier is modelled to generate probability scores over a set of classes. Hence, we use cross-entropy loss to train the attack classifier.

CHAPTER 6: EXPERIMENTS

We evaluate the performance of our network on AID under various settings. We also present extensive ablations that support the design choices for signature extractor and attack classifier.

Implementation details: The signature extractor comprises of 5 GLoF blocks with the attention arm embedding dimension of 768 and the convolutional arm embedding dimension of 64. The T2I block consists of two convolutional layers with kernel size 5 each followed by batch normalization. We use a patch size of 16x16 and 12 attention heads. Each convolutional arm in the GloF block consists of a ResNet block with 2 convolutional layers of kernel size 5, batch norm and a skip connection. We use DenseNet121[24] as the attack classifier. Final layers of the attack classifier are adjusted to compute probabilities over 13 classes for attack identification and 3 classes for attack family identification.

GLoF Variants: Standard GLoF block consists of convolution and attention arms. We introduce variants of GLoF block that exclusively contain either of the arms allowing us to study the contribution of local and global features separately. We term GLoF-C, referring to the GLoF block with only the convolutional arm and GLoF-A, referring to the GLoF block containing only the attention arm.

Experimental Setup: We employ a two stage training strategy to train the overall pipeline. In the first stage, the signature extractor is trained to produce the rectified image. Input adversarial images are randomly augmented by resizing and cropping. Benign samples corresponding to the adversarial inputs are used as the ground truth. Adam optimizer and L_2 loss are used to pre-train the signature extractor. In the second stage, the overall pipeline with the pre-trained signature extractor is further trained. We refrain from using any augmentations while training the complete pipeline since augmenting the adversarial samples results in alteration of the underlying perturbation thus

Table 6.1: **Performance of different methods on AID** focusing on identifying 13 different attacks and 3 attack families.

Method	**Attack Identification**	**Attack Family Identification**	**no. of params**
ResNet50[23]	68.27%	80.11%	24.7M
ResNet101[23]	71.03%	80.38%	43.8M
ResNet152[23]	67.03%	78.48%	59.5M
DenseNet121[24]	73.20%	84.21%	8.2M
DenseNet169[24]	72.22%	84.10%	14.3M
DenseNet201[24]	73.07%	81.69%	20.2M
InceptionV3[51]	69.96%	81.91%	22.9M
ViT-B/16[18]	63.91%	75.89%	85.8M
ViT-B/32[18]	54.61%	72.34%	87.4M
ViT-L/16[18]	67.28%	78.25%	303M
ViT-L/32[18]	55.23%	72.62%	305M
Ours	**80.14%**	**84.72%**	47.8M

making it difficult to comprehend the patterns responsible in identifying the attack. We use cross-entropy loss to train the network with Adam optimizer with a learning rate of $1e^{-4}$ and momentum rates of 0.9 and 0.999. We use exponential decay strategy to decrease the learning rate by 5% every 1k iterations. All experiments are conducted on NVIDIA V100 GPU with a batch size of 16. Two stage training helps in faster convergence of the overall network, allows the signature extractor to learn better, and removes the need to retrain it if novel attacks are included.

Baselines: Since the RED problem is first-of-its-kind, and there is no existing literature directly related to this problem, we develop several baselines and compare our technique against them. RED at its core is a classification problem, we look at the existing visual classifier models and train them accordingly for the RED problem. We consider variants of ResNet [23], DenseNet [24], Inception [51] and different versions of Vision Transformer[18]-$\{$ViT-B, ViT-L$\}$as baselines. In line with [18], ViT-B refers to the Base version of ViT with 12 encoder layers and ViT-L refers to the Large version with 24 encoder layers. We analyze patch sizes of 16x16 and 32x32 for both the

Table 6.2: **Cross Model Attack Identification.** AID-R, AID-D, and AID-I refer to the subsets of AID containing perturbations corresponding to the target models ResNet50[23], DenseNet121[24] and InceptionV3[51] respectively.

Method	**Train Set**	**Performance on different test sets**		
		AID-R	**AID-D**	**AID-I**
ResNet50 [23]	AID-R	71.46%	65.74%	62.90%
	AID-D	66.15%	66.88%	61.46%
	AID-I	59.69%	65.22%	66.96%
DenseNet121 [24]	AID-R	70.01%	66.89%	58.46%
	AID-D	55.77%	73.71%	53.83%
	AID-I	63.30%	66.96%	69.54%
InceptionV3 [51]	AID-R	66.35%	60.51%	61.29%
	AID-D	63.02%	66.05%	62.54%
	AID-I	59.21%	60.03%	68.72%
Proposed Approach	AID-R	**75.41%**	73.56%	69.76%
	AID-D	70.46%	**74.42%**	67.42%
	AID-I	69.95%	69.88%	**73.12%**

variants.

Results

Attack Identification: Table 6.1 reports the results on RED problem under two settings: identifying the attack as well as the attack family. Our approach with the pre-trained signature extractor, feature fusion and the attack classifier achieves an accuracy of **80.14%** on the attack identification and **84.72%** on attack family identification.

Comparison with baselines: Table 6.1 compares the performance of our network against other baselines. The top performing compared method, DenseNet121 [24], is surpassed by our technique in both categories by a margin of 6.94% in attack identification and 0.51% in attack family identification. In general, variants of ResNet [23] and Inception [51] under perform when compared with DenseNet [24] versions. Comparing with versions of ViT, CNNs have fewer number of parameters and perform much better in both the settings. One reason for this being that ViT[18] requires large amounts of training data. We also observe a drop in accuracy with increase in a patch size from 16x16 to 32x32 suggesting that ViT[18] struggles to accurately capture the local intrinsic properties as the patch becomes bigger. It is evident that the RED problem focusing on identifying the attack family is simpler compared to identifying the specific attack. This is in view of the fact that attacks that belong to the same family employ similar training strategies making it difficult to distinguish them.

Cross Model attack identification: We analyze the performance of our network on cross model attack identification. AID consists of attacks generated by targeting 3 different networks. For this experiment, we split AID into three subsets containing perturbations related to the corresponding target model. AID-R, AID-D, AID-I refer to subsets of AID containing perturbations corresponding to ResNet50 [23], DenseNet121 [24] and InceptionV3 [51] as target networks. Each subset is further split into train and test sets. Table 6.2 summarizes the results on cross model attack identification of several baselines compared against our technique. In general, we observe that the networks perform well when trained and tested on the same subsets of AID. The proposed technique performs better in all cases compared to other baselines. This experiment suggests that the although perturbations differ with the target model, the attacker leaves traces that can be leveraged to profile the attacker.

Ablations

We investigate the contribution of each component by performing ablation studies, summarized in Table 6.3a. At the core of signature extractor is the GLoF block. Removing local or global

Table 6.3: Performance of the proposed network and their variants

(a) Ablation study for Attack Identification.

Method	**Accuracy**
Full model	**80.14%**
without pre-training	79.20%
without global connect.- GLoF-C	78.66%
without local connect.- GLof-A	73.61%
without Feature Fusion	78.87%
without Signature Extractor	73.20%

(b) Evaluation of GLoF with varying number (m) of attention heads.

GLoF Variant	**PSNR**	**SSIM**
GLoF-C	31.49	0.88
GLoF-A	31.53	0.87
GloF($m = 4$)	30.96	0.87
GloF($m = 8$)	30.93	0.88
GloF($m = 12$)	**31.55**	**0.89**
GloF($m = 16$)	31.54	**0.89**

connectivity reduces the accuracy by 6.53% and 1.49% respectively. The extracted signature is coupled with the features from the input image. Removing feature fusion drops the accuracy of the network to 78.87%. Transformers are known to work well when pre-trained [18, 11]. We test the capability of our network without pre-training the signature extractor and observe a drop of 0.94%. Lastly, we mention the attack identification accuracy is 73.20% without any signature extractor.

Table 6.4: Performance of the Signature extractor/Attack classifier and their variants

(a) Effect of number of GLoF blocks n on attack identification

Number of GLoF Blocks	**Accuracy**
$n = 1$	79.20%
$n = 3$	79.65%
$n = 5$	**80.14%**
$n = 7$	79.22%
$n = 9$	79.90%

(b) Performance of different attack classifiers with Signature Extractor

Method	**Accuracy**
SigExt. +	
ResNet50[23]	73.80%
ResNet101[23]	74.74%
ResNet152[23]	73.25%
DenseNet121[24]	**80.14%**
DenseNet169[24]	78.15%
DenseNet201[24]	76.69%
InceptionV3[51]	70.38%

In Table 6.4a, we analyze the performance of the network by **varying the number of GLoF blocks**. Signature Extractor with as low as a single GLoF block achieves 79.20% (+6% over baseline) thus indicating its effectiveness. Employing 5 GLoF blocks yields the best accuracy of 80.14%.

We explore how **different attack classifiers** affect the overall performance. As seen in Table 6.4b, the signature extractor paired with DenseNet121 [24] yields the best results. It can be observed that the attack classifiers paired with signature extractor (Table 6.4b) performs significantly better compared to training stand alone classifiers. This supports the claim that extracting input-specific signature form the adversarial input to identify the attack is a better strategy.

Analysis

Signature Extraction: We investigate the performance of pre-training the signature extractor under various settings. To measure the performance of the signature extractor we use PSNR and SSIM metrics over the rectified image and the corresponding benign samples to evaluate the quality of the reconstruction. Table 6.3b reports the results of different variants of the GLoF block. Standard GLoF achieves higher PSNR and SSIM scores over GLoF-C and GLoF-A indicating that global and local connectivity together help in better reconstruction. We also report the variation in reconstruction scores when the number of heads m in multi head attention are increased. GLoF blocks with 12 heads achieves the highest scores of **31.55** PSNR and **0.89** SSIM.

We analyze the performance of the network by **varying the number of GLoF blocks**. Signature Extractor with as low as a single GLoF block achieves 79.20% (+6% over baseline) thus indicating its effectiveness. Employing 5 GLoF blocks yields the best accuracy of 80.14%.

Next, we explore how **different attack classifiers** affect the overall performance. We observe

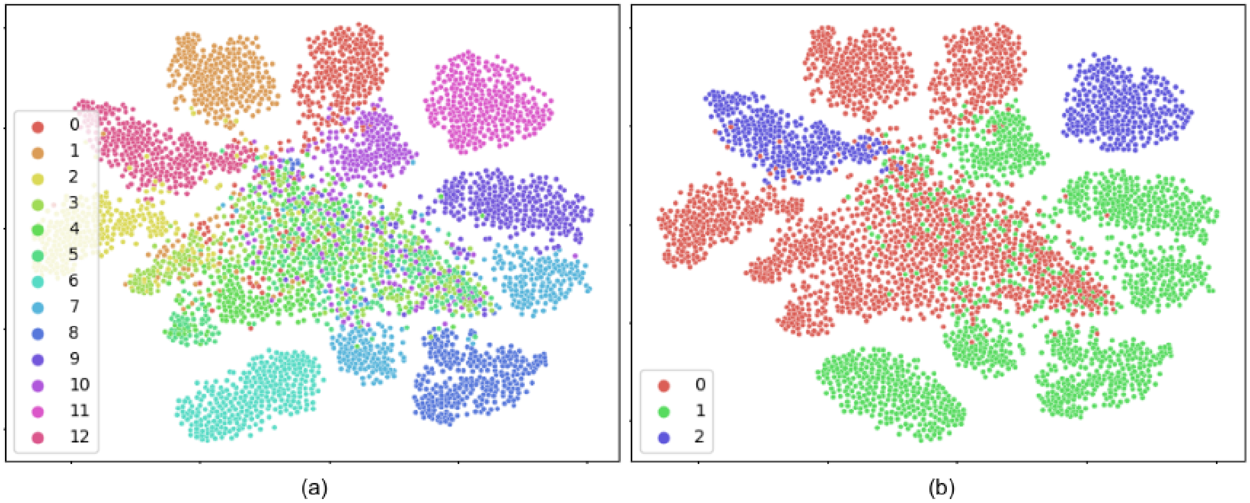
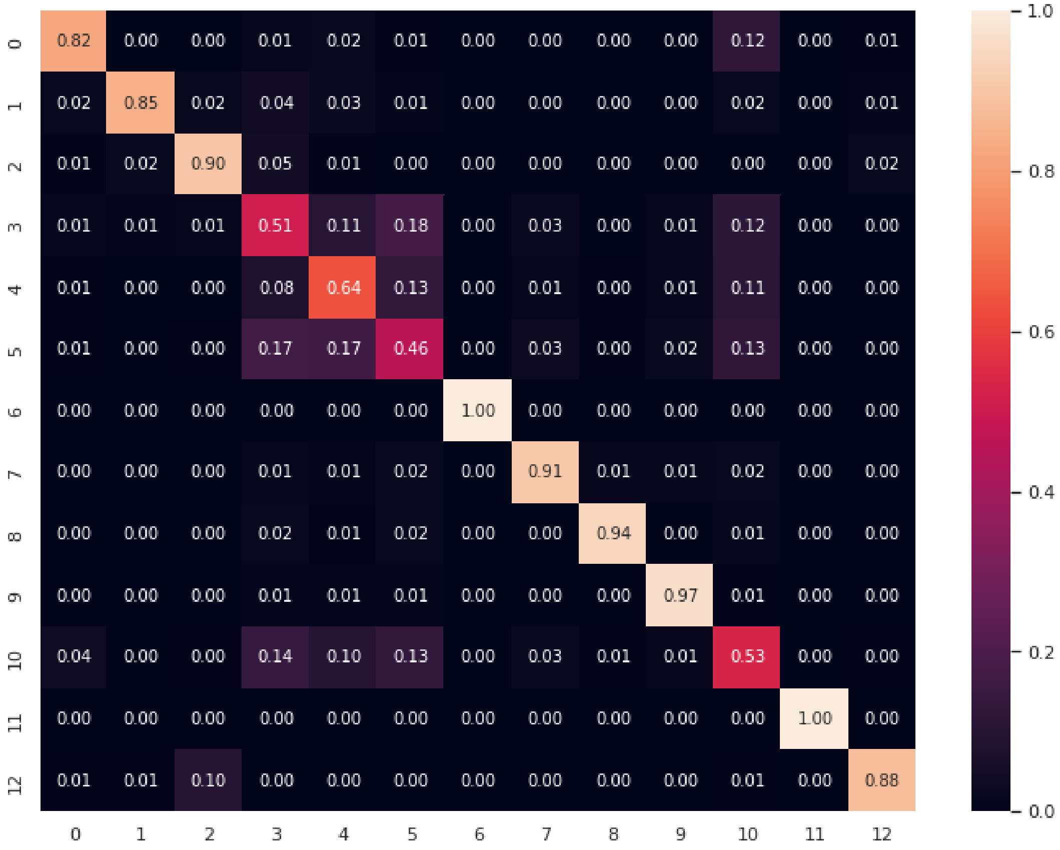


Figure 6.1: Visualizations of features learned by the attack classifier. (a)t-SNE for specific attack categories. Labels are in accordance with Table 4.2 (b)t-SNE for attack families. The labels {0,1,2} refer to {gradient, decision, universal} attacks.

that the versions of DenseNet121[24] perform better than other classifiers. The signature extractor paired with DenseNet121[24] yields the best results.

Identifying Novel Attacks: With the increasing threat to deep learning networks, it is highly likely for the RED problem to encounter novel unseen attacks. To experiment the effectiveness of the proposed network we devise an experiment which includes identifying the toolchain family of a novel attack. For this, we split AID into two different sets containing mutually exclusive attack categories. We retrain the overall pipeline on one set and test it on the novel classes which achieves an accuracy of 57.2%. We extend our approach to register novel attacks with minimal training set using toolchain indexing(discussed in supplementary). Identifying open set novel attacks under RED scenario remains challenging due to the fact that the unseen perturbations are nearly imperceptible and are difficult to distinguish.

Visualization: We generate t-SNE plots of a set of features extracted from the penultimate layer of the attack classifier. Fig.6.1 shows the three toolchain families forming separate clusters. Due



	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0.82	0.00	0.00	0.01	0.02	0.01	0.00	0.00	0.00	0.00	0.12	0.00	0.01
1	0.02	0.85	0.02	0.04	0.03	0.01	0.00	0.00	0.00	0.00	0.02	0.00	0.01
2	0.01	0.02	0.90	0.05	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02
3	0.01	0.01	0.01	0.51	0.11	0.18	0.00	0.03	0.00	0.01	0.12	0.00	0.00
4	0.01	0.00	0.00	0.08	0.64	0.13	0.00	0.01	0.00	0.01	0.11	0.00	0.00
5	0.01	0.00	0.00	0.17	0.17	0.46	0.00	0.03	0.00	0.02	0.13	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.01	0.01	0.02	0.00	0.91	0.01	0.01	0.02	0.00	0.00
8	0.00	0.00	0.00	0.02	0.01	0.02	0.00	0.00	0.94	0.00	0.01	0.00	0.00
9	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.97	0.01	0.00	0.00
10	0.04	0.00	0.00	0.14	0.10	0.13	0.00	0.03	0.01	0.01	0.53	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
12	0.01	0.01	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.88

Figure 6.2: **Confusion matrix:** The labels of the classes are in accordance with the order in Table 6.5.

to their ’universality’ constraint, universal perturbations form a clear cluster and are easily distinguishable. While gradient based attacks share similar techniques, decision based attacks have distinctive approaches based on the decision of the network. Hence we observe the overlap between gradient and decision based attacks. Fig shows the t-SNE plots over specific classes. Boundary Attack[6] has the maximum overlap with other attacks. In gradient based attacks, DeepFool[38], NewtonFool[26] and CW[8] attacks overlap with each other indicating that they generate similar patterns thus making it difficult to distinguish them.

We analyze class wise scores and the confusion matrix of the predictions from the proposed approach in Fig 6.2 and Table 6.5. From the confusion matrix, we observe the common trend of relatively high scores for all decision based attacks except for boundary attack. With scores close to 1, these attacks have distinctive patterns which are being easily identified by the signature extractor. Boundary attack do not always have specific patterns because of the way they are generated. Starting from a point that is already adversarial, boundary attack performs a random walk on the decision boundary minimizing the amount of perturbation. Similarly, universal attacks generate discernible patterns making it easier for detection. Major confusion occurs in the gradient based attacks among NewtonFool, DeepFool and CW attack. These attacks being highly powerful, are targeted on generated nearly imperceptible perturbations specific to the input image, making it difficult for the method to identify and distinguish. Similar trends observed in the confusion matrix can be seen in Table 6.5.

Reconstructions

Fig **??**. depicts the adversarial images, corresponding perturbations and the signatures extracted by the signature extractor. In general, the extracted signatures have patterns highlighting the object from the clean image. This is due to the fact that extracting these nearly imperceptible perturbations accurately always is almost nearly impossible. These patterns along with the patterns pertaining to the attacker help in training the attack classifier to identify the attacker.

Detecting Clean vs. Perturbed

While the core idea of RED is to profile the attacker given an adversarial image, it is likely for the signature extractor to be tested with clean images in real world scenarios. We devise an experiment to analyze the performance of the signature extractor in distinguishing clean from perturbed images. We consider a subset of AID containing adversarial images and similar size set of clean

Table 6.5: Classification report(Precision, Recall, F1-score) of the proposed network on AID.

label	Attack Method	Precision	Recall	F1-score
0	PGD [37]	0.90	0.82	0.86
1	BIM [30]	0.95	0.85	0.89
2	FGSM [20]	0.86	0.90	0.88
3	DeepFool [38]	0.49	0.51	0.50
4	NewtonFool [26]	0.59	0.64	0.61
5	CW [8]	0.48	0.46	0.47
6	Additive Gaussian [47]	1.00	1.00	1.00
7	Gaussian Blur [47]	0.90	0.91	0.90
8	Salt&Pepper [47]	0.97	0.94	0.95
9	Contrast Reduction [47]	0.92	0.97	0.94
10	Boundary [6]	0.49	0.53	0.51
11	UAN [22]	1.00	1.00	1.00
12	UAP [40]	0.95	0.88	0.91

images. The signature extractor is trained to extract signatures highlighting the patterns in adversarial images. Extracted signatures are used to train a binary classifier that identifies clean and perturbed images. We use a standard ResNet50[23] as the binary classifier in this case. The end to end pipeline yields a **100%** accuracy in distinguishing perturbed images from clean images. This can act as a preliminary step, and if a perturbation is detected, it can be passed to the attack classifier for identifying the specific attack category.

Enrolling Novel Classes

With the fast moving field of adversarial machine learning, it is highly likely for the signature extractor to come across novel unseen attacks. While, it is difficult to retrain the signature extractor and the attack classifier each time a new attack is added to the system, we employ a dictionary based toolchain indexing scheme to enrol novel attack classes with limited data.

We use a simple indexing scheme which can work as an addition to the existing signature extraction approach. The extracted signature for the adversarial images is of size 24x224x3. Since, storing and indexing such large images requires large amounts of memory and computations, we project the signature to a 512-dimensional using the model activations. These are extracted from the penultimate layer of a standard DenseNet-121 network. To index these compressed signatures into a dictionary, it is required to assign the correct toolchain to the sample. To solve this problem, we adapt a sparse and collaborative representation based classifier[3]. This classifier expects training data, which is our dictionary and the test sample that we need to index in the dictionary. The produced label is the label of the adversarial attack in our case, which is identifiable because our dictionary is structured. We use Orthogonal Matching Pursuit(OMP)[44] algorithm to compute sparse codes for a given sample over a fixed dictionary. It tends to assign large coefficient values in the sparse codes corresponding to the dictionary elements that are closely correlated to the test samples. The algorithm does not make any assumption about the dictionary itself. Hence, it does not restrict us from enrolling new attacks (or their families) to the dictionary.

To enroll a novel attack, we adopt a similar strategy. The main challenge for the indexing scheme is to register the novel attack with limited data. Hence, the indexing challenge gets translated into maintaining reasonable classification performance with one or very few samples for the unknown class. For analysis, we sequentially consider each of the thirteen attacks as the ‘unknown attack’ and note the performance of indexing scheme with varied number of samples available from the known attacks. We consider a subset of 50 samples per class from AID for the experiment. Starting with enrolling as little as a single sample for the class, we analyze the performance when we have 10, 20, 30, 40 and 50 samples for a newly enrolled class. The corresponding plots are shown in Fig 6.3.

From the plots, it can be observed for PGD, BIM and FGSM the indexing technique achieves accuracies greater than 60% with just 10 training samples. With as low as a single training sample, accuracy is consistently above 30%. For relatively simple classes like Gaussian Blur and UAP, we were able to maintain 100%. For particularly challenging classes like NewtonFool, CW and UAN, more samples resulted in better performance. These results demonstrate that the degradation in performance of our indexing scheme in the case of fewer training samples is graceful, to the extent that average accuracy across all classes with a single training sample is 46%. Hence, we can claim that the scheme has the ability of enrolling new attack effectively with as little as a single sample for most of the unknown attacks.

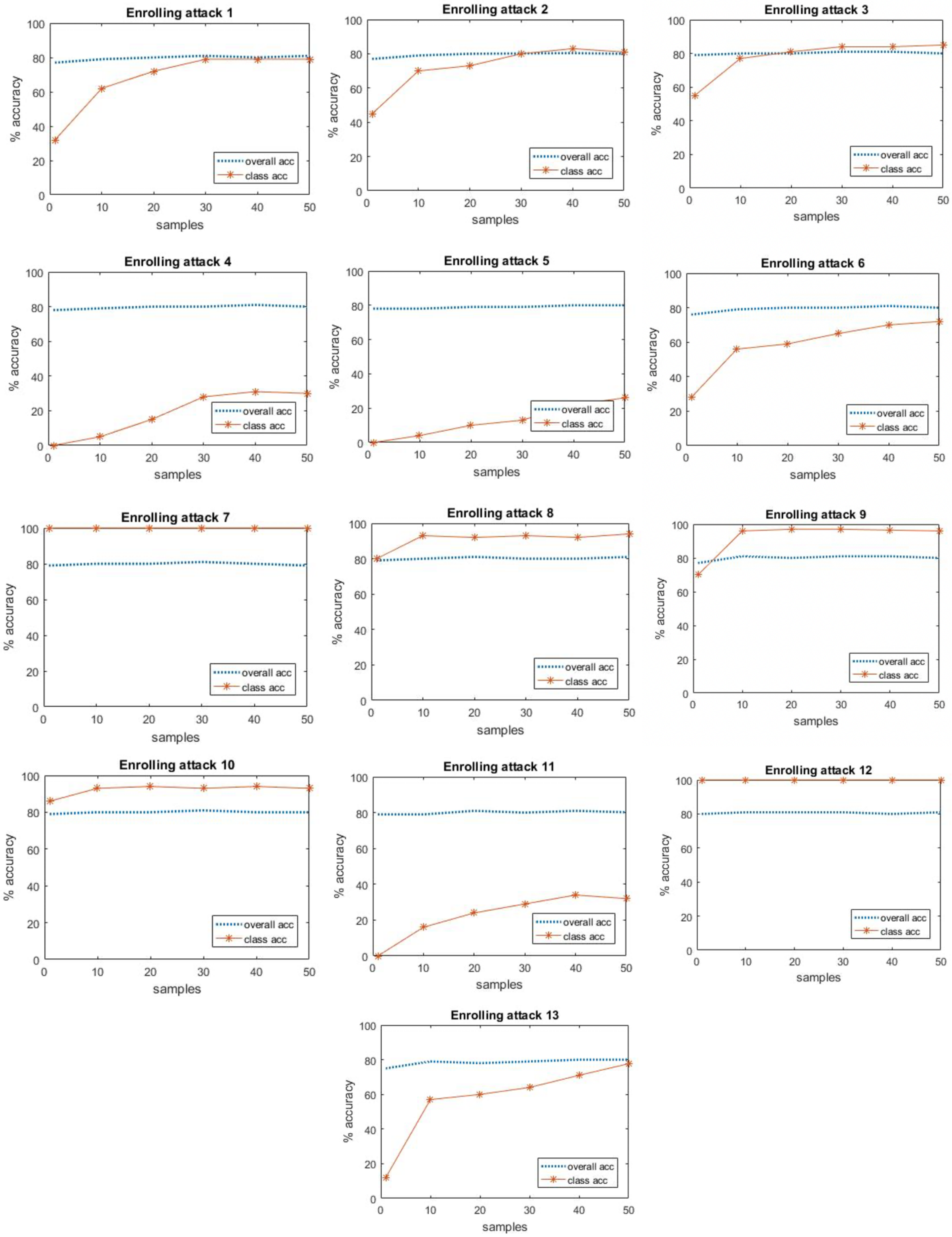


Figure 6.3: Results of considering individual classes as the unknown class

CHAPTER 7: CONCLUSION

We presented a new perspective on adversarial attacks indicating the presence of peculiar patterns in the perturbations that hint back to the attacker. We formulate the RED problem- given the adversarial input, reverse engineer to identify the attack strategy leveraged to generate the sample. We develop Adversarial Identification Dataset (AID) and compare several baseline techniques. Targeting RED, we propose a framework that combines CNN's capability to capture local features and transformer's ability to attain global attention together to generate a signature containing attack specific patterns, which is used by the attack classifier to identify the attacker. Extensive experiments showcase the efficacy of the proposed framework and support the credibility of RED problem.

LIST OF REFERENCES

[1] Akhtar, N., Mian, A.: Threat of adversarial attacks on deep learning in computer vision: A survey. Ieee Access **6**, 14410–14430 (2018)

[2] Akhtar, N., Mian, A., Kardan, N., Mubarak, S.: Advances in adversarial attacks and defenses in computer vision: A survey. arXiv preprint arXiv:2108.00401v2 (2021)

[3] Akhtar, N., Shafait, F., Mian, A.: Efficient classification with sparsity augmented collaborative representation. Pattern Recognition **65**, 136–145 (2017)

[4] Andriushchenko, M., Croce, F., Flammarion, N., Hein, M.: Square attack: a query-efficient black-box adversarial attack via random search (2020)

[5] Bai, Y., Zeng, Y., Jiang, Y., Wang, Y., Xia, S.T., Guo, W.: Improving query efficiency of black-box adversarial attack. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 101–116. Springer International Publishing, Cham (2020)

[6] Brendel, W., Rauber, J., Bethge, M.: Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In: International Conference on Learning Representations (2018), `https://openreview.net/forum?id=SyZI0GWCZ`

[7] Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 ieee symposium on security and privacy (sp). pp. 39–57. IEEE (2017)

[8] Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (SP). pp. 39–57 (2017). https://doi.org/10.1109/SP.2017.49

[9] Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A., Mukhopadhyay, D.: Adversarial attacks and defences: A survey (2018)

[10] Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A., Mukhopadhyay, D.: A survey on adversarial attacks and defences. CAAI Transactions on Intelligence Technology **6**(1), 25–45 (2021)

[11] Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., Gao, W.: Pre-trained image processing transformer (2021)

[12] Chen, W., Zhang, Z., Hu, X., Wu, B.: Boosting decision-based black-box adversarial attacks with random sign flip. In: ECCV (2020)

[13] Croce, F., Hein, M.: Provable robustness against all adversarial l_p-perturbations for $p \geq 1$. In: International Conference on Learning Representations (2020), `https://openreview.net/forum?id=rklk_ySYPB`

[14] Deng, Z., Yang, X., Xu, S., Su, H., Zhu, J.: Libre: A practical bayesian approach to adversarial detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 972–982 (2021)

[15] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL (2019)

[16] Din, S.U., Akhtar, N., Younis, S., Shafait, F., Mansoor, A., Shafique, M.: Steganographic universal adversarial perturbations. Pattern Recognition Letters **135**, 146–152 (2020)

[17] Ding, G.W., Wang, L., Jin, X.: AdverTorch v0.1: An adversarial robustness toolbox based on pytorch. arXiv preprint arXiv:1902.07623 (2019)

[18] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. ICLR (2021)

[19] Du, J., Zhang, H., Zhou, J.T., Yang, Y., Feng, J.: Query-efficient meta attack to deep neural networks. arXiv preprint arXiv:1906.02398 (2019)

[20] Goodfellow, I., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)

[21] Guo, C., Rana, M., Cisse, M., van der Maaten, L.: Countering adversarial images using input transformations. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=SyJ7ClWCb

[22] Hayes, J., Danezis, G.: Learning universal adversarial perturbations with generative models. In: 2018 IEEE Security and Privacy Workshops (SPW). pp. 43–49. IEEE Computer Society, Los Alamitos, CA, USA (may 2018). https://doi.org/10.1109/SPW.2018.00015, https://doi.ieeecomputersociety.org/10.1109/SPW.2018.00015

[23] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

[24] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)

[25] Huang, Z., Zhang, T.: Black-box adversarial attack with transferable model-based embedding. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=SJxhNTNYwB

[26] Jang, U., Wu, X., Jha, S.: Objective metrics and gradient descent algorithms for adversarial examples in machine learning. In: Proceedings of the 33rd Annual Computer Security Applications Conference. p. 262–277. ACSAC 2017, Association for Computing Ma-

chinery, New York, NY, USA (2017). https://doi.org/10.1145/3134600.3134635, https://doi.org/10.1145/3134600.3134635

[27] Kannan, H., Kurakin, A., Goodfellow, I.: Adversarial logit pairing. arXiv preprint arXiv:1803.06373 (2018)

[28] Katz, G., Barrett, C., Dill, D.L., Julian, K., Kochenderfer, M.J.: Reluplex: An efficient smt solver for verifying deep neural networks. In: International Conference on Computer Aided Verification. pp. 97–117. Springer (2017)

[29] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25**, 1097–1105 (2012)

[30] Kurakin, A., Goodfellow, I., Bengio, S., et al.: Adversarial examples in the physical world (2016)

[31] LeCun, Y., Bengio, Y., et al.: Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks **3361**(10), 1995 (1995)

[32] Li, H., Xu, X., Zhang, X., Yang, S., Li, B.: Qeba: Query-efficient boundary-based blackbox attack. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1221–1230 (2020)

[33] Li, S., Zhu, S., Paul, S., Roy-Chowdhury, A.K., Song, C., Krishnamurthy, S.V., Swami, A., Chan, K.S.: Connecting the dots: Detecting adversarial perturbations using context inconsistency. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. (eds.) Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIII. Lecture Notes in Computer Science, vol. 12368, pp. 396–413. Springer (2020). https://doi.org/10.1007/978-3-030-58592-1_24, https://doi.org/10.1007/978-3-030-58592-1_24

[34] Li, Y., Zhang, K., Cao, J., Timofte, R., Van Gool, L.: Localvit: Bringing locality to vision transformers. arXiv preprint arXiv:2104.05707 (2021)

[35] Liao, F., Liang, M., Dong, Y., Pang, T., Hu, X., Zhu, J.: Defense against adversarial attacks using high-level representation guided denoiser. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1778–1787 (2018)

[36] Ma, C., Chen, L., Yong, J.H.: Simulating unknown target models for query-efficient black-box attacks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11835–11844 (June 2021)

[37] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)

[38] Moosavi-Dezfooli, S., Fawzi, A., Frossard, P.: Deepfool: A simple and accurate method to fool deep neural networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2574–2582. IEEE Computer Society, Los Alamitos, CA, USA (jun 2016). https://doi.org/10.1109/CVPR.2016.282, https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.282

[39] Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1765–1773 (2017)

[40] Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)

[41] Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2574–2582 (2016)

[42] Nicolae, M.I., Sinn, M., Tran, M.N., Buesser, B., Rawat, A., Wistuba, M., Zantedeschi, V., Baracaldo, N., Chen, B., Ludwig, H., Molloy, I., Edwards, B.: Adversarial robustness toolbox v1.2.0. CoRR **1807.01069** (2018), https://arxiv.org/pdf/1807.01069

[43] Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: 2016 IEEE European symposium on security and privacy (EuroS&P). pp. 372–387. IEEE (2016)

[44] Pati, Y.C., Rezaiifar, R., Krishnaprasad, P.S.: Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In: Proceedings of 27th Asilomar conference on signals, systems and computers. pp. 40–44. IEEE (1993)

[45] Qin, Y., Frosst, N., Sabour, S., Raffel, C., Cottrell, G., Hinton, G.: Detecting and diagnosing adversarial images with class-conditional capsule reconstructions. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=Skgy464Kvr

[46] Rahmati, A., Moosavi-Dezfooli, S.M., Frossard, P., Dai, H.: Geoda: a geometric framework for black-box adversarial attacks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8446–8455 (2020)

[47] Rauber, J., Brendel, W., Bethge, M.: Foolbox: A python toolbox to benchmark the robustness of machine learning models. In: Reliable Machine Learning in the Wild Workshop, 34th International Conference on Machine Learning (2017), http://arxiv.org/abs/1707.04131

[48] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y

[49] Shi, Y., Han, Y., Tian, Q.: Polishing decision-based adversarial noise with a customized sampling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1030–1038 (2020)

[50] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al.: Mastering the game of go without human knowledge. nature **550**(7676), 354–359 (2017)

[51] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)

[52] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)

[53] Tjeng, V., Xiao, K.Y., Tedrake, R.: Evaluating robustness of neural networks with mixed integer programming. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=HyGIdiRqtm

[54] Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., McDaniel, P.: Ensemble adversarial training: Attacks and defenses. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=rkZvSe-RZ

[55] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)

[56] Wang, H., Wu, Z., Liu, Z., Cai, H., Zhu, L., Gan, C., Han, S.: Hat: Hardware-aware transformers for efficient natural language processing. In: Annual Conference of the Association for Computational Linguistics (2020)

[57] Xie, C., Wang, J., Zhang, Z., Ren, Z., Yuille, A.: Mitigating adversarial effects through randomization. In: International Conference on Learning Representations (2018)

[58] Xie, C., Wu, Y., Maaten, L.v.d., Yuille, A.L., He, K.: Feature denoising for improving adversarial robustness. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 501–509 (2019)

[59] Xu, W., Evans, D., Qi, Y.: Feature squeezing: Detecting adversarial examples in deep neural networks. arXiv preprint arXiv:1704.01155 (2017)

[60] Yuan, K., Guo, S., Liu, Z., Zhou, A., Yu, F., Wu, W.: Incorporating convolution designs into visual transformers (2021)

[61] Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., Jordan, M.: Theoretically principled trade-off between robustness and accuracy. In: International Conference on Machine Learning. pp. 7472–7482. PMLR (2019)

[62] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of AAAI (2021)