This ICCV workshop paper is the Open Access version, provided by the Computer Vision Foundation. Except for this watermark, it is identical to the accepted version; the final published version of the proceedings is available on IEEE Xplore.

PRAT: PRofiling Adversarial aTtacks

Rahul Ambati[1] Naveed Akhtar[2] Ajmal Mian[2] Yogesh S Rawat[1]
[1]University of Central Florida [2]The University of Western Australia
rahul.ambati@knights.ucf.edu {naveed.akhtar,ajmal.mian}@uwa.edu.au yogesh@crcv.ucf.edu

Abstract

Intrinsic susceptibility of deep learning to adversarial examples has led to a plethora of attack techniques with a broad common 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 'PRofiling Adversarial aTtacks' (PRAT). Given an adversarial example, the objective of PRAT 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, which we also study. To enable PRAT 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 PRAT objective. Our framework utilizes a Transformer based Global-LOcal Feature (GLOF) module to extract an approximate signature of the adversarial attack, which in turn is used for the identification of the attack. Using AID and our framework, we provide multiple interesting benchmark results for the PRAT problem. The dataset and the code are available at https://github.com/rahulambati/PRAT

1. Introduction

Deep learning is currently at the center of many emerging technologies, from autonomous vehicles to numerous security applications. However, it is also well-established that deep networks are susceptible to adversarial attacks [1, 8]. This intriguing weakness of deep learning, which is otherwise known to supersede human intelligence in complex tasks [41], has attracted an ever-increasing interest of the research community in the last few years [9]. This has led to a wide range of adversarial attacks that can effectively fool deep learning. Although adversarial attacks have also led to research in defenses, there is a consensus that defenses currently lack efficacy. Many of them are easily broken, or become ineffective by changing the attack strategy [2].

Figure 1: Despite their imperceptibility, adversarial perturbations contain peculiar patterns. Perturbations generated using the popular methods FGSM, DeepFool, CW and UAP Attacks are shown.

Incidentally, deep learning in practice is still widely open to malicious manipulation through adversarial attacks [8]. It is yet to be seen if this technology can retain its impressive performance while also demonstrating robustness to adversarial attacks. Until an 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 is almost completely void of any exploration along this line. From the pragmatic viewpoint, the primal question of this potential research 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 PRofiling Adversarial aTtacks (PRAT). Focusing on the *additive adversarial perturbations*, our aim is to explore the extent to which a victim is able to identify its attacker by analyzing only the 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. AID covers input-specific and input-agnostic attacks and considers white-box and black-box setups. We select these attacks considering the objective of retracing the attacker from the adversarial image.

We use AID to explore PRAT 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, these traces might 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. We design a signature extractor consisting Global-LOcal Feature extractor (GLOF) modules that combine CNN's ability to learn local structure [26] and transformer's capability to capture global information [46, 47, 14]. These signatures contain information which corresponds to the attack algorithm and we use this signature to identify the attack leveraged to generate the adversarial example.

Our contributions are summarized as follow.

- We put forth a new problem of PRofiling Adversarial aTtacks (PRAT), aimed at profiling the attacker. We formalize PRAT to provide a systematic guideline for research in this direction.
- We propose an effective framework to provide the first-of-its-kind solution to the PRAT problem which consists of a hybrid Transformer network that combines the capabilities of CNNs and attention networks targeted to solve PRAT.
- We introduce a large Adversarial Identification Dataset (AID), comprising 180k+ adversarial samples generated with 13 different attacks. AID is used to extensively study PRAT, leading to promising results.

2. Related Work

Adversarial attacks and defenses are currently a highly active research direction. 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 [43]. [43] demonstrated that deep models can be fooled into incorrect prediction by adding imperceptible adversarial perturbations to the input. Hence, to efficiently compute adversarial samples (for adversarial training), [16] 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.

[25] 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 [31], 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 hence categorized as white-box attacks. Other popular gradient based attacks include Carlini & Wagner attack [6], DeepFool [34] and Jacobian Saliency Map Attack (JSMA) [35].

Black-box attacks do not assume any knowledge of the model, except its predictions. The most popular streams 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 feedback for improving the adversarial nature of the input. If the attacker only receives the model decision (not its confidence score), then such an attack is called a decision-based attack. Currently, the decision based attacks are more popular in black-box setups due to their pragmatic nature. A few recent representative examples in this category include [37], [40], [15], [27].

With the discovery of adversarial samples, there is an increased interest in devising defences, of which, the most popular strategy is adversarial training [16, 22, 31, 45, 48].

The existing literature also covers a wide range of other defense techniques, from augmenting the models with external defense modules [36, 28, 12] to certified defenses [23, 44, 11]. Here, we emphasize that these defenses generally come at considerable computational cost and degradation in model performance on clean inputs.

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.

3. The PRAT Problem

The PRofiling Adversasrial aTtacks (PRAT) 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

$$\mathcal{C}(\mathbf{I} + \boldsymbol{\rho}) \rightarrow \tilde{\boldsymbol{\ell}} \text{ s.t. } \tilde{\boldsymbol{\ell}} \neq \boldsymbol{\ell}, ||\boldsymbol{\rho}||_p < \eta. \quad (1)$$

The above formulation underpins the most widely adopted settings for the adversarial attacks, where $\boldsymbol{\rho}$ is a systemat-

ically computed additive signal. From our PRAT 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.

In practice, the targeted model $\mathcal{C}$ must already be deployed and the input $\mathbf{I}$ fixed during an attack leaving$\mathcal{A}$ as the point of interest for the PRAT 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. (1) are overseen by the elements of $\{\boldsymbol{\psi}\}$. Collectively, both sets contain all the hyper-parameters available to an attacker to compute $\boldsymbol{\rho}$.

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 profiling. These three elements are 1) φ_1*-model gradient information*, 2) φ_2*-black-box prediction score information*, and 3) 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` for a family of attacks $\mathcal{F}_1^a$ of gradient-based attacks and can be `false` for non-gradient based attack family $\mathcal{F}_1^b$. Similarly, when $\varphi_2 =$ `true`, we get an attack family $\mathcal{F}_2^a$ of score-based black-box attacks[30, 20], and $\varphi_2 =$ `false` yields $\mathcal{F}_2^b$ that represents decision-based attacks[4, 3, 10]. Similarly, $\varphi_3 =$ `true` results in the $\mathcal{F}_3^a$ representing universal attacks and $\varphi_3 =$ `false` corresponds to the family $\mathcal{F}_3^b$ consisting input-specific attacks.

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.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. (1). Then, $f_z^i(\mathbf{I}) \rightarrow \tilde{\mathbf{I}}$ s.t. $\mathcal{C}(\tilde{\mathbf{I}}) \rightarrow \tilde{\ell} \neq \ell, ||\boldsymbol{\rho}||_p < \eta$. In this setting, the core PRAT 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.

4. Adversarial Identification Dataset (AID)

To investigate the PRAT problem, we develop Adversarial Identification Dataset (AID). Below, we detail different attacks $\mathcal{A}$, attack families $\mathcal{F}$ and their design and numeric hyper-parameters $(\{\boldsymbol{\varphi}\}, \{\boldsymbol{\psi}\})$ considered in AID.

Most of the existing literature on adversarial attacks concentrates on devising novel attack schemes or robustifying models against the attacks. Multiple existing adversarial attack libraries 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. In line with our PRAT problem, AID consists of 3 different attack families (*gradient-based*, *decision-based*, and *universal*) with 13 different attack strategies resulting in over 180k samples. We discuss these families next.

4.1. Attack Families

Gradient based attacks: Gradient based attacks 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)* [16], *Basic Iterative Method* [25], *Newton-Fool* [21], *Projected Gradient Descent(PGD)* [31], *DeepFool* [32], *Carlini Wagner (CW)* [7] attacks.

Decision based attacks: Decision-based attacks are 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*, *Gaussian Blur*, *Salt & Pepper Noise*, *Contrast Reduction*, and *Boundary Attack* [5] for this family.

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 modeling to compute image agnostic perturbations. We consider *Universal Adversarial Perturbation (UAP)* [33], *Universal Adversarial Network (UAN)* [17] for the universal attack family.

4.2. Dataset creation

Benign samples: We require clean images to create an adversarial perturbation. We utilize ImageNet2012 [39] validation set consisting of 50k images spanning across 1000 classes. We split the validation set into two exclusive parts, forming training and test partitions of AID. The 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 target models; ResNet50 [18], DenseNet121 [19] and InceptionV3 [42]. Using multiple models ensures that the adversarial samples are not model specific.

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. The procedure of generating the entire dataset as well as the summary statistics are further detailed in the supplementary material of the paper. We also summarise the considered attacks, their families, and used perturbation norm-bounds in Table 1.

5. Proposed Approach

Here, we discuss the design choices we consider for solving the PRAT problem $\Psi(\tilde{\mathbf{I}}) \rightarrow f_z^i$. A simple approach to solve PRAT 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 ρ are closely intertwined with the benign sample $\mathbf{I}$, thus making the problem much harder. To solve it, we design a signature extractor $\Omega(\tilde{\mathbf{I}}) \rightarrow \tilde{\boldsymbol{\rho}}$ that generates a signature $\tilde{\boldsymbol{\rho}}$ from the adversarial input s.t. it lies close to the original perturbation $\boldsymbol{\rho}$ while preserving patterns helpful in identifying the attacker. The objective of the signature extractor is

$$\Omega(\tilde{\mathbf{I}}) \rightarrow \tilde{\boldsymbol{\rho}}, \;\; ||\tilde{\boldsymbol{\rho}} - \boldsymbol{\rho}||_2 = \boldsymbol{\delta}, \;\; min(\boldsymbol{\delta}). \tag{2}$$

While the objective draws similarities with existing problems like denoising/deraining, signature extraction is relatively complex. Noise/rain pertaining to these tasks are largely localized in nature and are visually perceptible in most cases which is not the case for PRAT that makes the problem more challenging and requires methods beyond standard techniques aimed at denoising and other low-level computer vision tasks.

Extracted signature is utilized to train a classifier C that identifies the attack. The objective of the classifier is

$$C(\tilde{\boldsymbol{\rho}}) \rightarrow f_z^i, \;\; where \;\; f_z^i(\mathbf{I}) \rightarrow \tilde{\mathbf{I}}, \tag{3}$$

where, $\tilde{\boldsymbol{\rho}}$ is the generated signature, f_z^i is the z^{th} attack from the i^{th} toolchain family. Figure 2 shows an overview of the proposed approach highlighting the signature extractor and the attack classifier.

label	Attack Method	Family	Setup	NB
0	PGD [31]	Grad.	WB	l_∞
1	BIM [25]	Grad.	WB	l_∞
2	FGSM [16]	Grad.	WB	l_∞
3	DeepFool [32]	Grad.	WB	l_∞
4	NewtonFool [21]	Grad.	WB	l_2
5	CW [7]	Grad.	WB	l_2
6	Additive Gaussian [38]	Grad.	BB	l_2
7	Gaussian Blur [38]	Grad.	BB	l_∞
8	Salt&Pepper [38]	Grad.	BB	l_∞
9	Contrast Reduction [38]	Dec.	BB	l_∞
10	Boundary [5]	Dec.	BB	l_2
11	UAN [17]	Uni.	WB	l_∞
12	UAP [33]	Uni.	WB	l_∞

Table 1: **Summary of the attacks in AID**. Grad., Dec. and Uni. denote Gradient-based, Decision-based and Universal attacks. BB and WB denote Black- and White-box attacks. NB is the norm bound on perturbation.

Signature Extractor: It serves the purpose of extracting a signature with patterns specific to the attack. As shown in Fig.2, the signature extractor has two streams of information flow progressing through a series of GLOF modules. Each stream is designed to capture local or global features along with feature sharing across them. GLOF module utilizes convolutional layers to extract local features while attention mechanism applied over image patches 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 adversarial image yields the signature.

The input adversarial image $\tilde{\boldsymbol{I}} \in \mathbb{R}^{H \times W \times 3}$ (H, W correspond to image height and width and 3 corresponds to the RGB channels) is split into a series of patches. The patches are flattened and projected onto the embedding space of dimension $\boldsymbol{D_1}$. Similar to [14], we add positional embeddings to the patch embeddings. The resulting patch embedding is termed $\boldsymbol{T_0} \in \mathbb{R}^{N \times D_1}$ (0 referring to the initial feature level and N referring to the number of patches).

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 refers to the initial feature level). Features extracted from previous level $(l-1)$ are passed on to the next GLOF module.

$$\boldsymbol{T_l}, \boldsymbol{Z_l} = GLOF(\boldsymbol{T_{l-1}}, \boldsymbol{Z_{l-1}}); \quad l = 1...L \tag{4}$$

Where L is the number of GLOF modules. The output of the final GLOF module 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 $\tilde{\boldsymbol{\rho}} = \tilde{\boldsymbol{I}} - \boldsymbol{I_r}$.

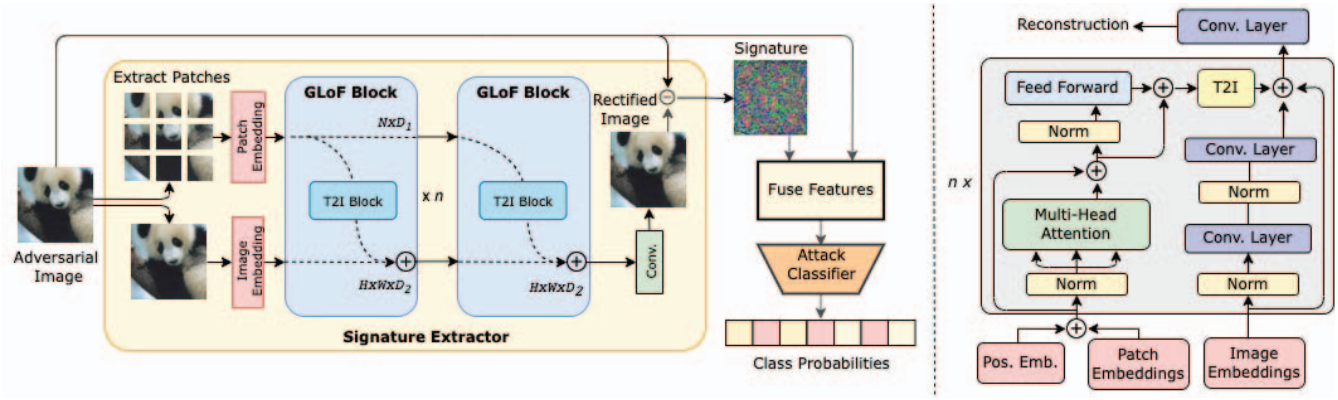

Figure 2: (**Left**) Schematics of the proposed approach. (**Right**) GLoF module architecture. In our method, an input adversarial image is passed through a series of GLoF modules. Each GLoF module has two arms; one captures global dependencies, the other captures local features. Extracted signature is fused with the adversarial image and fed to attack classifier.

GLOF Module: Standard convolutional layers are good at extracting local patterns [24]. On the other hand, transformers are known to be extremely powerful in learning non-local connectivity [13]. As seen in [14], vision transformers fail to utilize the local information [29]. 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 module is given in Fig.2.

The GLOF module at any level receives the local and global features from the previous level.

Local features: Embedded 2D image features from the previous layer $\boldsymbol{Z_{l-1}}$ are fed to a ResNet block[18] with convolutional, batch norm and activation layers.

Global features: Embedded tokens are fed to attention mechanism[46]. 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 consisting of two dense layers that are applied to individual tokens separately with GELU applied over the output of the first dense layer[14].

T2I Block: Features from the attention arm corresponding to the global connectivity are merged with the convolutional arm. **Token to Image (T2I)** is responsible for rearranging the series of tokens to form a 2D grid. This transformed grid is passed 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 module. The merged features as well as the learned token embeddings are passed to consecutive GLoF modules.

Attack Classifier: The generated signature is specific to the input. Since the input contains contextual information, 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 separately and concatenating them.

Training objective: We utilize two learning objectives in our framework. We use L_2 loss to minimize the distance of generated signature $\tilde{\boldsymbol{\rho}}$ to the raw perturbation $\boldsymbol{\rho}$. Alongside, the attack classifier is modelled with cross-entropy loss to generate probability scores over a set of classes.

6. Experiments

We evaluate the performance of the proposed approach on AID under various settings and also present extensive ablations that support the design choices.

Implementation details: The signature extractor comprises of 5 GLOF modules 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 module consists of a ResNet block with 2 convolutional layers of kernel size 5, batch norm and a skip connection. We use DenseNet121[19] 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 module consists of convolution and attention arms. We introduce variants of GLOF 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 module with only the convolutional arm and GLOF-A, referring to the GLOF module 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. 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,

Method	Attack Identification	Attack Family Identification	no. of params
ResNet50[18]	68.27%	80.11%	24.7M
ResNet101[18]	71.03%	80.38%	43.8M
ResNet152[18]	67.03%	78.48%	59.5M
DenseNet121[19]	73.20%	84.21%	8.2M
DenseNet169[19]	72.22%	84.10%	14.3M
DenseNet201[19]	73.07%	81.69%	20.2M
InceptionV3[42]	69.96%	81.91%	22.9M
ViT-B/16[14]	63.91%	75.89%	85.8M
ViT-B/32[14]	54.61%	72.34%	87.4M
ViT-L/16[14]	67.28%	78.25%	303M
ViT-L/32[14]	55.23%	72.62%	305M
Ours	**80.14%**	**84.72%**	47.8M

Table 2: **Performance of different methods on AID** focusing on identifying 13 different attacks and 3 attack families.

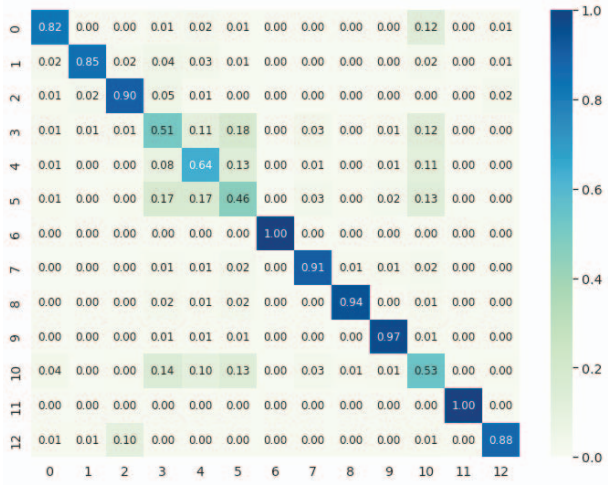

Figure 3: **Confusion matrix:** The labels of the classes are in accordance with the order in Table 1.

the overall pipeline with the pre-trained signature extractor is further trained. 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.

Evaluation metrics: Since the main objective of the PRAT is classification, we use accuracy to compare across several techniques. We also evaluate the performance of the signature extractor using PSNR and SSIM scores calculated over the rectified image and the benign sample.

Baselines: Since the PRAT problem is first-of-its-kind, we develop several baselines and compare our technique against them. PRAT at its core is a classification problem, we look at the existing visual classifier models and train them accordingly for the PRAT problem. We consider variants of ResNet [18], DenseNet [19], Inception [42] and different versions of Vision Transformer[14]-{ViT-B, ViT-L}as baselines. In line with the original work, ViT-B refers to the Base version of ViT with 12 encoder layers and ViT-L is the Large version with 24 encoder layers. We analyze patch sizes of 16x16 and 32x32 for both the variants.

6.1. Results

Attack Identification: Table 2 reports the results on PRAT problem evaluated on AID 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 **80.14%** accuracy on the attack identification and **84.72%** on attack family identification.

Comparison with baselines: Table 2 compares the performance of our network against other baselines. The top performing compared method, DenseNet121[19], 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[18] and Inception[42] under perform when compared with DenseNet[19] versions. Comparing with versions of ViT[14], CNNs have fewer number of parameters and perform much better in both the settings. One reason for this being that ViT 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[14] struggles to accurately capture the local intrinsic properties as the patch becomes bigger. We also evaluate the performance of Wiener filtering combined with a classifier. This setting achieves 67.55% compared to 80.14% by the GLOF based model. It is evident that identifying attack family is simpler compared to identifying the specific attack.

6.2. Ablations

Table 3 presents the ablation study on the proposed network. Full model refers to the complete pipeline with pre-trained signature extractor and a classifier accepting fused features form the signature and the input which yields 80.14% on AID.

Effect of pre-training: Transformers are known to work well with pre-training. Without pre-training of the signature extractor the accuracy drop to 79.20%.

Effect of GLOF module: Signature extractor with exclusively GLOF-C variant yields an accuracy of 78.66% while its counter part with GLOF-A variant(without CNN blocks) only achieves 73.61% indicating the importance of both the components for a better performance on PRAT.

Effect of Fusion: The fusion module combines the features from the extracted signature and the adversarial image. Removing such fusion module and only relying on the

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

Table 3: **Ablation study for Attack Identification.** Full model has a pre-trained signature extractor and a classifier accepting fused product of the signature and features.

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

Table 4: **Quantitative results of Signature Extractor**. GLoF-C and GLoF-A refer to the variants of GLoF exclusively containing local and global connectivity respectively.

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

Table 5: Effect of number of GLoF modules n on the performance of attack identification

extracted signature results in an accuracy of 78.87%.

Effect of Signature Extractor: While Signature Extractor acts as the crux of the overall pipeline, removing it is no different than the baseline DenseNet121 [12] from table 2 which yields 73.20%.

6.3. Analysis and Discussion

Confusion Matrix: We analyze class wise scores and the confusion matrix of the predictions from the proposed approach in Fig 3. 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. 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.

Method	Train Set	Performance on different test sets		
		AID-R	AID-D	AID-I
ResNet50 [18]	AID-R	71.46%	65.74%	62.90%
	AID-D	66.15%	66.88%	61.46%
	AID-I	59.69%	65.22%	66.96%
DenNet121 [19]	AID-R	70.01%	66.89%	58.46%
	AID-D	55.77%	73.71%	53.83%
	AID-I	63.3%	66.96%	69.54%
InceptionV3 [42]	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%**

Table 6: **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[18], DenseNet121[19] (abbreviated as DenNet121) and InceptionV3[42] respectively.

Signature Extraction: Table 4 investigates the performance of the signature extractor under various settings. Standard GLOF achieves higher PSNR and SSIM scores over GLOF-C and GLOF-A indicating that global and local connectivity used in conjunction 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 modules with 12 heads achieves the highest scores of **31.55** PSNR and **0.89** SSIM.

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

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[18], DenseNet121[19] and InceptionV3[42] as target networks. Each subset is further split into train-test sets. Table 6 details 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 perturbations from different target models also contain similar

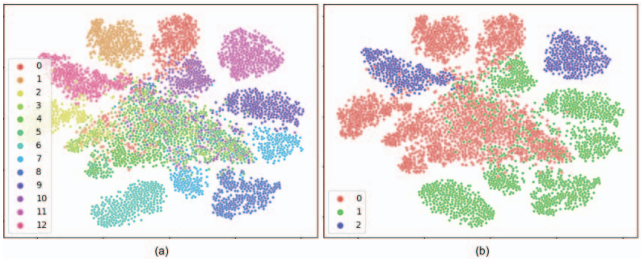

Figure 4: **Visualizations of features learned by the attack classifier.** (a) t-SNE for specific attack categories. Labels are according to Table 1 (b) t-SNE for attack families. Labels {0,1,2} refer to {gradient, decision, universal} attacks.

traces that can be leveraged to profile the attacker.

Success rate vs. Identifiability: While the stronger attacks like PGD have a 100% fooling rate, the weaker black box attacks have a success rate of atleast 65% for the samples considered in AID. We also study the indentifiability vs success rate for the FGSM class and find that our technique achieves 74.9% accuracy for an epsilon as low as 2 and 94.5% for an epsilon of value 16. We observe an upward increasing trend as the epsilon increases indicating an increasing level of perceptibility of the patterns.

Identifying unseen attacks: With the increasing threat to neural networks, it is likely for the PRAT 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 an unseen attack. For this, we split AID into two different sets containing mutually exclusive attack categories. We retrain the overall pipeline and test it on the unseen 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 PRAT scenario remains challenging due to the fact that the unseen perturbations are nearly imperceptible and are difficult to distinguish.

Feature visualization: We study the separability of extracted features by analyzing the t-SNE plots of a set of features extracted from the penultimate layer of the attack classifier. Fig.4 shows the three toolchain families forming separate clusters. Due 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.4 shows the t-SNE plots over specific classes. Boundary Attack has the maximum overlap with other attacks. In gradient based attacks, DeepFool, NewtonFool and CW attacks overlap with each other indicating that they generate similar patterns thus making it difficult to distinguish them.

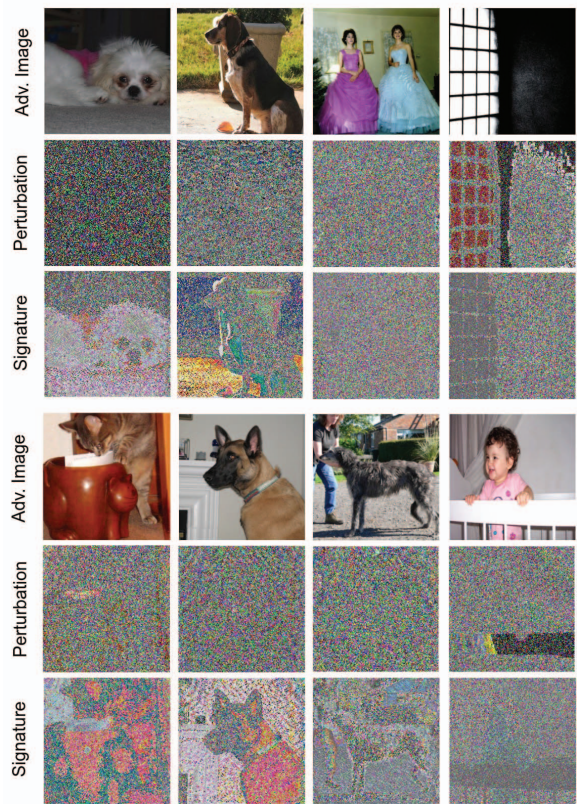

Figure 5: Adversarial images, their perturbations (normalized) and the corresponding signature(normalized) extracted by the proposed approach.

Reconstructions: Fig 5. depicts the adversarial images, corresponding perturbations and the signatures extracted by the our method. In general, the extracted signatures have patterns highlighting the object from the image. This is due to the fact that extracting these nearly imperceptible perturbations accurately is always challenging. These patterns along with the patterns pertaining to the attacker help in training the attack classifier to identify the attacker.

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 PRAT problem - given the adversarial input, profile the attack signature to identify the attack used to generate the sample. We develop Adversarial Identification Dataset and compare several baseline techniques on the proposed dataset. Targeting PRAT, we propose a framework that combines CNN's capability to capture local features and Transformer's ability to encode global attention to generate signatures containing attack-specific patterns, which are used by an attack classifier to identify the attack. Extensive experiments showcase the efficacy of our framework and support the credibility of the proposed PRAT problem.

References

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

[2] Naveed Akhtar, Ajmal Mian, Navid Kardan, and Shah Mubarak. Advances in adversarial attacks and defenses in computer vision: A survey. *arXiv preprint arXiv:2108.00401v2*, 2021. 1

[3] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. 2020. 3

[4] Yang Bai, Yuyuan Zeng, Yong Jiang, Yisen Wang, Shu-Tao Xia, and Weiwei Guo. Improving query efficiency of black-box adversarial attack. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 101–116, Cham, 2020. Springer International Publishing. 3

[5] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations*, 2018. 3, 4

[6] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017. 2

[7] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017. 3, 4

[8] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*, 2018. 1

[9] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. A survey on adversarial attacks and defences. *CAAI Transactions on Intelligence Technology*, 6(1):25–45, 2021. 1

[10] Weilun Chen, Zhaoxiang Zhang, Xiaolin Hu, and Baoyuan Wu. Boosting decision-based black-box adversarial attacks with random sign flip. In *ECCV*, 2020. 3

[11] Francesco Croce and Matthias Hein. Provable robustness against all adversarial l_p-perturbations for $p \geq 1$. In *International Conference on Learning Representations*, 2020. 2

[12] Zhijie Deng, Xiao Yang, Shizhen Xu, Hang Su, and Jun Zhu. Libre: A practical bayesian approach to adversarial detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 972–982, 2021. 2, 7

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 5

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 2, 4, 5, 6

[15] Jiawei Du, Hu Zhang, Joey Tianyi Zhou, Yi Yang, and Jiashi Feng. Query-efficient meta attack to deep neural networks. *arXiv preprint arXiv:1906.02398*, 2019. 2

[16] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 2, 3, 4

[17] J. Hayes and G. Danezis. Learning universal adversarial perturbations with generative models. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 43–49, Los Alamitos, CA, USA, may 2018. IEEE Computer Society. 3, 4

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4, 5, 6, 7

[19] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 4, 5, 6, 7

[20] Zhichao Huang and Tong Zhang. Black-box adversarial attack with transferable model-based embedding. In *International Conference on Learning Representations*, 2020. 3

[21] Uyeong Jang, Xi Wu, and Somesh Jha. Objective metrics and gradient descent algorithms for adversarial examples in machine learning. In *Proceedings of the 33rd Annual Computer Security Applications Conference*, ACSAC 2017, page 262–277, New York, NY, USA, 2017. Association for Computing Machinery. 3, 4

[22] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018. 2

[23] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer, 2017. 2

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 5

[25] Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world, 2016. 2, 3, 4

[26] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995. 2

[27] Huichen Li, Xiaojun Xu, Xiaolu Zhang, Shuang Yang, and Bo Li. Qeba: Query-efficient boundary-based blackbox attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1221–1230, 2020. 2

[28] Shasha Li, Shitong Zhu, Sudipta Paul, Amit K. Roy-Chowdhury, Chengyu Song, Srikanth V. Krishnamurthy, Ananthram Swami, and Kevin S. Chan. Connecting the dots: Detecting adversarial perturbations using context inconsistency. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIII*, volume 12368 of *Lecture*

Notes in Computer Science, pages 396–413. Springer, 2020. 2

[29] Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021. 5

[30] Chen Ma, Li Chen, and Jun-Hai Yong. Simulating unknown target models for query-efficient black-box attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11835–11844, June 2021. 3

[31] 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. 2, 3, 4

[32] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582, Los Alamitos, CA, USA, jun 2016. IEEE Computer Society. 3, 4

[33] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3, 4

[34] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016. 2

[35] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016. 2

[36] Yao Qin, Nicholas Frosst, Sara Sabour, Colin Raffel, Garrison Cottrell, and Geoffrey Hinton. Detecting and diagnosing adversarial images with class-conditional capsule reconstructions. In *International Conference on Learning Representations*, 2020. 2

[37] Ali Rahmati, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Huaiyu Dai. Geoda: a geometric framework for black-box adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8446–8455, 2020. 2

[38] Jonas Rauber, Wieland Brendel, and Matthias Bethge. 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. 4

[39] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 3

[40] Yucheng Shi, Yahong Han, and Qi Tian. Polishing decision-based adversarial noise with a customized sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1030–1038, 2020. 2

[41] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017. 1

[42] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 4, 6, 7

[43] 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. 2

[44] Vincent Tjeng, Kai Y. Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. In *International Conference on Learning Representations*, 2019. 2

[45] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018. 2

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2, 5

[47] Hanrui Wang, Zhanghao Wu, Zhijian Liu, Han Cai, Ligeng Zhu, Chuang Gan, and Song Han. Hat: Hardware-aware transformers for efficient natural language processing. In *Annual Conference of the Association for Computational Linguistics*, 2020. 2

[48] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pages 7472–7482. PMLR, 2019. 2