Efficient and Robust Classification for Sparse Attacks

Mark Beliaev[1], Payam Delgosha[2], Hamed Hassani[3], and Ramtin Pedarsani[4]

2022 IEEE International Symposium on Information Theory (ISIT) | 978-1-6654-2159-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISIT50566.2022.9834832

***Abstract*—In the past two decades we have seen the popularity of neural networks increase in conjunction with their classification accuracy. Parallel to this, we have also witnessed how fragile the very same prediction models are: tiny perturbations to the inputs can cause misclassification errors throughout entire datasets. In this paper, we consider perturbations bounded by the ℓ_0–norm, which have been shown as effective attacks in the domains of image-recognition, natural language processing, and malware-detection. To this end, we propose a novel defense method that consists of "truncation" and "adversarial training". We then theoretically study the Gaussian mixture setting and prove the asymptotic optimality of our proposed classifier. Motivated by the insights we obtain, we extend these components to neural network classifiers. We conduct numerical experiments in the domain of computer vision using the MNIST and CIFAR datasets, demonstrating significant improvement for the robust classification error of neural networks.**

I. Introduction

Today we see machine learning at the heart of many safety-critical applications, including image recognition, autonomous driving, and virtual assistance. This comes with little surprise, as we have seen deep neural networks gain tremendous popularity due to their success, showing near human performance in the image-recognition domain [1], as well as successful application in natural language processing [2], and playing games [3], [4]. Instead, what is surprising is how fragile these neural networks are when subjected to adversarial attacks.

Adversarial attacks are methods that try to fool prediction models by adding small perturbations to their inputs. They were initially shown to be effective in causing classification errors throughout different machine learning models [5]–[7]. Following this, a lot of effort has been put into generating increasingly more complex attack models that can utilize a small amount of semantic-preserving modifications, while still being able to fool a classifier [8]–[10]. Typically, this is done by constraining the perturbations with an ℓ_p–norm, where the most common settings use either ℓ_∞ [8], [9], [11]–[15], ℓ_2 [9], [16]–[19], or ℓ_1 [20], [21]. As of now, the state-of-the-art empirical defense against adversarial attacks is iteratively retraining with adversarial examples [8]. While adversarial retraining by itself can help improve robustness, we have seen a fundamental trade-off between robustness and clean accuracy, as well as a lack of generalization across different attacks [22]–[26].

In this paper we focus on a different setting, where adversarial perturbations are constrained by the ℓ_0–norm. This setting has gained considerable attention [9], [21], [27]–[30] due to applications in object detection [31], [32] and NLP [33]. In these applications, robust guarantees against ℓ_0–attacks are specifically important since there is an inherent limit on the number of input features that can be modified. In the previously described settings, the adversary was able to modify all of the elements of the input, while still satisfying the given constraint. Conversely, in the ℓ_0 setting the adversary is given a budget k, and is directly constrained to perturbing at most k coordinates within the input. In other words, the adversary is allowed to change the input within the ℓ_0–ball of radius k, where k is typically much smaller than the input dimension, and hence the name *sparse attacks*. In addition, unlike ℓ_p–balls ($p \geq 1$), the ℓ_0–ball has a more complex geometry: it is non-convex, highly non-smooth, and unbounded. In combination with these properties, the ℓ_0–ball's inherent discrete structure provides fundamental challenges that are absent in other adversarial settings studied in the literature, making most techniques from prior work non-applicable. Crucially, piece-wise linear classifier, e.g. neural networks with ReLU activations, were shown to fail in this setting [34], where recent work has demonstrated the ability ℓ_0–attacks have in confusing image classifiers [9], [10], [27], [28], [35]. Thus, our current architecture designs and learning procedures have to be rethought based on the unique geometry of the ℓ_0–norm. We set out to accomplish this goal in this paper.

Two notable works have proposed defenses against the related but less powerful $(\ell_0 + \ell_\infty)$–adversary: the *Analysis by Synthesis* (ABS) model [28] and randomized ablation [30]. Here the adversary is also constrained by the number of coordinates it can perturb, but these perturbations can no longer be arbitrarily large due to the bound posed by the ℓ_∞–norm on the value that each coordinate can take. Although the proposed defenses show improved robustness guarantees when classifying the MNIST and CIFAR datasets, we see these guarantees vanish as the ℓ_∞–bound is relaxed, while our method is able to generalize to both settings (more details are provided in Table II located in Section V). On top of this, we note that the aforementioned defenses rely on computationally expensive solutions.

Building on our prior work [36], we develop an algorithm that directly tackles the ℓ_0 setting, and prove that in the Gaussian mixture setting we can achieve asymptotic optimality. In our prior work [36] we showed that in order to achieve robustness against sparse attacks, we need two novel and non-linear components, namely *truncation* and *filtration*. However, filtration turns out to be computationally expensive, hence in this paper we replace filtration with adversarial training and prove that in the Gaussian mixture setting, truncation combined with adversarial training results in a classifier which is asymptotically optimal. In other words, we propose a

We thank NSF grant #2003035 and NSF grant #1909320.
[1]Uni. of California Santa Barbara, mbeliaev@ucsb.edu
[2]Uni. of Illinois at Urbana-Champaign, delgosha@illinois.edu
[3]Uni. of Pennsylvania, hassani@seas.upenn.edu
[4]Uni. of California Santa Barbara, ramtin@ucsb.edu

practical classifier which achieves the best possible robust classification error in the presence of an ℓ_0 adversary in a certain asymptotic setting. The effectiveness of our proposed method is validated through empirical study. More precisely, utilizing the state-of-the-art sparse attack of `sparse-rs` [29] as well as the commonly used `Pointwise Attack` [28], we show that while adversarial training without truncation fails in robustifying against ℓ_0–attacks, our method has strong performance both in terms of robustness and computational efficiency when tested on the MNIST [37] and CIFAR [38] datasets.

II. Problem Setup

We consider the general M–class classification problem, where given an input $\boldsymbol{x} \in \mathbb{R}^d$ and its label $y \in \{1, \dots, M\}$, we aim to construct a model that can accurately predict the label given the input. We can think of the input and labels as coming from some distribution $(\boldsymbol{x}, y) \in \mathcal{D}$, with our classifier belonging to the family of functions $\mathcal{C} : \mathbb{R}^d \mapsto \{1, \dots, M\}$. As a metric for the discrepancy between the label and the classifier's prediction for a given input $\boldsymbol{x}$, we use the $0-1$ loss $\ell(\mathcal{C}; \boldsymbol{x}, y) = \mathbb{1}[\mathcal{C}(\boldsymbol{x}) \neq y]$.

Given this setup we can introduce an ℓ_0–adversary, which perturbs the input $\boldsymbol{x}$ within the ℓ_0–ball of radius k: $\mathcal{B}_0(\boldsymbol{x}, k) := \{\boldsymbol{x}' \in \mathbb{R}^d : \|\boldsymbol{x} - \boldsymbol{x}'\|_0 \leq k\}$, where we define $\|\boldsymbol{x}\|_0 := \sum_{i=1}^d \mathbb{1}[x_i \neq 0]$ for $\boldsymbol{x} = (x_1, \dots, x_d)$, and refer to k as the *budget* of the adversary. This states that the adversary is allowed to arbitrarily modify at most k coordinates of $\boldsymbol{x}$ to obtain $\boldsymbol{x}'$, feeding the new vector $\boldsymbol{x}'$ to the classifier. Within this scope, the *robust classification error* of a classifier $\mathcal{C}$ is defined by:

$$\mathcal{L}_{\mathcal{D}}(\mathcal{C}, k) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\max_{\boldsymbol{x}' \in \mathcal{B}_0(\boldsymbol{x},k)} \ell(\mathcal{C}; \boldsymbol{x}', y)\right], \tag{1}$$

where we aim to design classifiers with the minimum *robust classification error*. To this end, we can define the *optimal robust classification error* as the result of minimizing (1) over all possible classifiers:

$$\mathcal{L}^*_{\mathcal{D}}(k) := \inf_{\mathcal{C}} \mathcal{L}_{\mathcal{D}}(\mathcal{C}, k). \tag{2}$$

Our goal is to characterize $\mathcal{L}^*_{\mathcal{D}}(k)$ as a function of the adversary's budget k. Specifically, we aim to find robust classifiers whose performance is close to the optimal robust classification error. Due to the complex geometry of the ℓ_0–ball, this poses a challenging problem. In fact, we have already seen how all conventional classifiers fail in this setting [34]. In order to address this problem, our current architecture designs and learning procedures have thus to be *rethought* based on the geometry of the perturbation set. To this end, we note that directly solving the optimization problem in (1) and finding the optimal robust error is intractable. Instead, inspired by robust statistics [39], we introduce truncation (see Section III-A) as the main building block of our classifier. We then aim to find the best robust classifier in the set of truncated classifiers. We prove in Section IV that this results achieves near-optimal robust classifiers in the Gaussian mixture setting. Furthermore, in Section III-C we discuss how we go beyond this Gaussian setting and use adversarial training to find the best truncated classifier in the general deep learning scenario.

III. The Proposed Algorithm

In this section we will go over the proposed algorithm, introducing how *truncation* is defined, followed by an explanation of how it can be extended to *fully connected* layers found within neural networks. We then describe the adversarial training component of our framework. As we will show in our theoretical and experimental results, coupling truncation with adversarial training is crucial to robustifying classifiers against ℓ_0–attacks. We defer the explanation of applying truncation to convolutional networks to Section V, where we discuss our experiments using the CIFAR dataset.

A. Truncation

Given $\boldsymbol{w}, \boldsymbol{x} \in \mathbb{R}^d$ and an integer $0 \leq k \leq d/2$, we define the *k–truncated inner product* of $\boldsymbol{w}$ and $\boldsymbol{x}$ as the summation of the element-wise product of $\boldsymbol{w}$ and $\boldsymbol{x}$ after removing the top and bottom k elements, and denote it by $\langle \boldsymbol{w}, \boldsymbol{x} \rangle_k$. If we define $\boldsymbol{u} := \boldsymbol{w} \odot \boldsymbol{x} \in \mathbb{R}^d$ as the element-wise product of $\boldsymbol{w}$ and $\boldsymbol{x}$, then letting $\boldsymbol{s} = (s_1, \dots s_n) = \text{sort}(\boldsymbol{u})$ be the result obtained after sorting $\boldsymbol{u}$ in descending order, we can define

$$\langle \boldsymbol{w}, \boldsymbol{x} \rangle_k := \sum_{i=k+1}^{d-k} s_i. \tag{3}$$

Note that when $k = 0$, the truncation operation in (3) reduces to the normal inner product denoted by $\langle \boldsymbol{w}, \boldsymbol{x} \rangle$. We can see that truncation is a natural method by which one can remove "outliers" found in the data after an adversary has modified some coordinates. Since an ℓ_0–adversary with a budget of k can modify at most k of the input's coordinates by an arbitrary amount, we can expect the k–truncated inner product to be robust against these ℓ_0 perturbations. In fact, we formalize this result in Section IV and show that truncation can be directly used to construct the optimally robust classifier in the setting of Gaussian mixture models attacked by an ℓ_0–adversary. Until then, we will focus the discussion on how we use truncation to construct robust neural networks.

To test the usability of the proposed truncation operator, we must consider how it can be applied within typical neural network architectures to improve their robustness. Within the scope of our notation in Section II, we restrict the family of classifiers $\mathcal{C} : \mathbb{R}^d \mapsto \{1, \dots, M\}$ to functions that can be represented by feed-forward neural networks composed of fully connected (FC) layers and non-linearities.

We denote a *fully connected feed-forward neural network with L layers* as a function $F(\boldsymbol{x}; \boldsymbol{\theta}) = y$ parameterized by $\boldsymbol{\theta}$, which takes an input $\boldsymbol{x} \in \mathbb{R}^d$, and returns the predicted label $y \in \{1, \dots, M\}$. This network can be viewed as a composite of L functions, referred to as layers, with non-linearities applied between the layers:

$$F(\boldsymbol{x}; \boldsymbol{\theta}) = \sigma_L(\boldsymbol{W}_L \sigma_{L-1}(\boldsymbol{W}_{L-1} \dots \sigma_1(\boldsymbol{W}_1 \boldsymbol{x}) \dots)), \tag{4}$$

where the parameters are $\boldsymbol{\theta} = (\boldsymbol{W}_1, \dots, \boldsymbol{W}_L)$ with $\boldsymbol{W}_l \in \mathbb{R}^{d_l \times d_{l-1}}$ and $d_0 = d$, and the non-linearities are $\sigma_l : \mathbb{R}^{d_l} \mapsto$

$\mathbb{R}^{d_l}$. In our work we use the well known ReLU [40] activation function for all of our non-linearities other than the one at the output layer σ_L, which is implemented as a softmax so that our function outputs a probability vector. Also note that we have left out denoting the bias terms added within the FC layers, as this can be taken care of by appending a constant coordinate to the input.

B. Robust Fully Connected Networks

We can naturally extend truncation to FC layers by defining this operation to act on a weight matrix $\boldsymbol{W}$ as such:

$$\langle \boldsymbol{W}, \boldsymbol{x} \rangle_k := \boldsymbol{u}, \text{ where } u_i := \langle \boldsymbol{W}[i], \boldsymbol{x} \rangle_k, \tag{5}$$

using $\boldsymbol{W}[i]$ to denote the i'th row of the weight matrix $\boldsymbol{W}$. Note that (5) returns a vector $\boldsymbol{u}$, whose i'th entry u_i is the result of applying our truncation operation shown in (3) on the row $\boldsymbol{W}[i]$ and vector $\boldsymbol{x}$. To form our k–truncated fully connected network $F^{(k)}$, we replace the first FC layer $\boldsymbol{W}_1\boldsymbol{x}$ in (4) with its k–truncated version defined in (5).

$$F^{(k)}(\boldsymbol{x}; \boldsymbol{\theta}) = \sigma_L(\boldsymbol{W}_L \sigma_{L-1}(\boldsymbol{W}_{L-1} \dots \sigma_1(\langle \boldsymbol{W}_1, \boldsymbol{x} \rangle_k) \dots)). \tag{6}$$

Note that with this formulation, $F^{(0)} = F$, since $\langle \boldsymbol{W}, \boldsymbol{x} \rangle_k = \boldsymbol{W}\boldsymbol{x}$ when $k = 0$. Applying truncation on the first layer ensures that the effect of the adversary is compensated at the early stages of the network and does not propagate through the layers.

C. Adversarial Training

Although truncation on its own is expected to increase a classifier's robustness, we suggest going further and coupling our framework with adversarial training as originally proposed by [8]. In the Gaussian mixture setting considered in Section IV, we prove that the asymptotically optimal classifier requires truncation, and to find its weights we need an optimization step that resembles adversarial training. We hypothesize that extending these theoretical results to neural networks will help improve their robustness, and we set out to improve the robust guarantees of a FC network F against an ℓ_0–attack with budget k. We accomplish this by turning F into its k–truncated counterpart $F^{(k)}$, and performing adversarial training (details provided in Section V) on $F^{(k)}$ by iteratively appending adversarial examples to the training data.

IV. Theoretical Framework

In this section, within the setup of Section II, we consider a Gaussian mixture setting and show that our algorithm achieves near optimal robust classification error, i.e., we show that the deviation from optimality is asymptotically vanishing. *The key insight that we obtain from our theoretical analysis is that truncation and adversarial training are the two major components that enable provable robustness against ℓ_0–attacks.*

More precisely, we consider the binary classification scenario where the distribution $\mathcal{D}$ is as follows. We have $y \in \{\pm 1\}$ with $\mathbb{P}(y = +1) = \mathbb{P}(y = -1) = 1/2$, and conditionally on y, we have $\boldsymbol{x} = y\boldsymbol{\mu} + \boldsymbol{z}$ where $\boldsymbol{\mu} \in \mathbb{R}^d$ and $\boldsymbol{z} \sim \mathcal{N}(0, \Sigma)$ is a Gaussian vector with zero mean and diagonal covariance matrix Σ. To simplify the discussion, we assume that Σ has strictly positive diagonal entries $\sigma_1^2, \dots, \sigma_d^2$. It is easy to verify that in the absence of the adversary, the optimal Bayes classifier is the linear classifier $\text{sgn}(\langle \boldsymbol{w}, \boldsymbol{x} \rangle)$ with $\boldsymbol{w} = \Sigma^{-1}\mu$. The corresponding optimal standard error of this classifier is $\bar{\Phi}(\|\Sigma^{-1/2}\boldsymbol{\mu}\|_2)$, where $\bar{\Phi}(.)$ denotes the complementary CDF of the standard normal distribution. Therefore, in order to fix the baseline, without loss of generality we assume that $\|\Sigma^{-1/2}\boldsymbol{\mu}\|_2 = 1$ so that the optimal standard error is $\bar{\Phi}(1)$.

Motivated by the fact that the optimal Bayes classifier in this setting is linear, we consider neural networks with a single layer. More precisely, we consider the family of k–truncated linear classifiers $\mathcal{C}_{\boldsymbol{w}}^{(k)} : \boldsymbol{x}' \mapsto \text{sgn}(\langle \boldsymbol{w}, \boldsymbol{x}' \rangle_k)$. Adopting our notation in (1), we denote the robust classification error of a classifier $\mathcal{C}_{\boldsymbol{w}}^{(k)}$ in this family by $\mathcal{L}_{\boldsymbol{\mu},\Sigma}(\mathcal{C}_{\boldsymbol{w}}^{(k)}, k)$. Moreover, as in (2), we denote the optimal robust classification error by $\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}(k)$. To simplify the notation, when the problem parameters $\boldsymbol{\mu}$ and Σ are clear from the context, we may remove them from the above notations and simply write $\mathcal{L}(\mathcal{C}_{\boldsymbol{w}}^{(k)}, k)$ and $\mathcal{L}^*(k)$.

A. Asymptotic Optimality of our Algorithm

To show that k–truncated linear classifiers are asymptotically optimal, we must first recall the results from our prior work which established a lower bound on the optimal robust classification by developing an attack strategy for the adversary and showing that no classifier can achieve better performance. We directly copy the result below for convenience:

Theorem 1 (Theorem 2 in [36])**.** *Assume that Σ is diagonal and let $\boldsymbol{\nu} = \Sigma^{-1/2}\boldsymbol{\mu}$. Then for any $A \subseteq \{1, \dots, d\}$, we have*

$$\mathcal{L}^*\left(\|\boldsymbol{\nu}_A\|_1 \log d\right) \geq \bar{\Phi}(\|\boldsymbol{\nu}_{A^c}\|_2) - \frac{1}{\log d},$$

where $\boldsymbol{\nu}_A$ and $\boldsymbol{\nu}_{A^c}$ denote the coordinates of $\boldsymbol{\nu}$ in the sets A and A^c, respectively.

As discussed in Section III, we use adversarial training in order to obtain the model weights, as it is indeed a proxy for optimizing $\boldsymbol{w}$ in the class of k–linear classifiers $\mathcal{C}_{\boldsymbol{w}}^{(k)}$. Letting $\boldsymbol{w}^*(k) \in \arg\min_{\boldsymbol{w}} \mathcal{L}(\mathcal{C}_{\boldsymbol{w}}^{(k)}, k)$, we show that the performance of $\mathcal{C}_{\boldsymbol{w}^*(k)}^{(k)}$ in the presence of an adversary with ℓ_0 budget k is comparable to the optimal robust classification error, with an asymptotically vanishing deviation.

To this end, given an error threshold $\bar{\Phi}(1) \leq \varepsilon \leq 1/2$ where ε ranges between the standard error $\bar{\Phi}(1)$ and the error corresponding to a random guess, we define $k^{\text{Trunc}}(\varepsilon) := \max\{k : \mathcal{L}(\mathcal{C}_{\boldsymbol{w}^*(k)}^{(k)}, k) \leq \varepsilon\}$ as the maximum adversarial budget that the *class of truncated linear classifiers* can tolerate to achieve a robust error of at most ε, with the truncation parameter chosen to be equal to adversary's budget. Defining $k^*(\varepsilon) := \max\{k : \mathcal{L}^*(k) \leq \varepsilon\}$ as the maximum adversarial budget that an *optimal classifier* can tolerate under the constraint of having a robust error of at most ε, we can see that $k^*(\varepsilon) \geq k^{\text{Trunc}}(\varepsilon)$.

As we will formally show below, k^{Trunc} and k^* are close to each other up to multiplicative factors that are sublinear in d.

In order to have a first order analysis and focus on the behavior of the adversary's budget as a power of the dimension d, we define $\alpha^{\text{Trunc}}(\varepsilon) := \log_d k^{\text{Trunc}}(\varepsilon)$ and $\alpha^*(\varepsilon) := \log_d k^*(\varepsilon)$. The following theorem shows that modulo some vanishing terms in d, α^{Trunc} is close to α^*. In other words, the class of linear truncation classifiers are asymptotically optimal for the above mixture Gaussian setting. Proof of Theorem 2 is provided in the full version of this paper [41].

Theorem 2. *Given* $\bar{\Phi}(1) + 1/\log d + \sqrt{2/\log d} < \varepsilon < \frac{1}{2}$, *there are constants* $c_i = c_i(\varepsilon, d)$, $i \in \{1, 2\}$, *which do not depend on the parameters of the problem (i.e.* $\boldsymbol{\mu}$ *and* Σ*) such that* $\lim_{d\to\infty} c_i(\varepsilon, d) = 0$ *for* $i \in \{1, 2\}$ *and*

$$\alpha^*(\varepsilon) \geq \alpha^{\textit{Trunc}}(\varepsilon) \geq \alpha^*(\varepsilon - c_1) - c_2.$$

Theorem 2 essentially says that up to asymptotically vanishing terms, the truncated classifier can tolerate as much adversarial budget as an optimal robust classifier. In order to prove this result, we use Theorem 1 which enables us to make sure that no other classifier can achieve better asymptotic performance, hence our algorithm is asymptotically optimal.

V. Experiments

To present our experimental results, we discuss (i) how we chose and modified the ℓ_0–attacks utilized in our experiments, and (ii) how under these modifications we saw the robust guarantees of prior work's previously proposed and well-studied ℓ_0–defense method vanish. Following this in V-A, we show how our k–truncated FC networks performed on MNIST, and propose a heuristically motivated extension of truncation to 2–dimensional convolution layers, testing it on CIFAR.

For our work, we mainly utilize `sparse-rs` [29], a sparse black-box l_0-attack framework. Given a pixel budget k, time budget t, input image $\boldsymbol{x}$, and a classifier $\mathcal{C}$, this attack performs a random search where it tries to change a set of k pixels in $\boldsymbol{x}$ that cause the new adversarial image $\boldsymbol{x}'$ to be misclassified by $\mathcal{C}$. The creators of `sparse-rs` have shown their framework outperforms all previous black- and white-box attacks, and hence we use this attack within our adversarial training framework and after training to approximately measure the *robust accuracy* of our classifier. We also utilize the `Pointwise Attack` [28] to directly compare our results with other ℓ_0-defense techniques [30]. This attack tries to greedily minimize the ℓ_0–norm by first adding salt-and-pepper noise, and then repeatedly resetting perturbed pixels while keeping the image misclassified. Since here we cannot directly control the number of allowed perturbations k, we only use this attack to measure the *median adversarial attack magnitude* as was done in prior work [30], denoting this value with ρ.

Before moving on, we point out that we normalize the coordinates of our inputs to be within some defined range $[-a, a]$. By design, the ℓ_0–attacks mentioned also require the *perturbed* coordinates to lie within some range $[-\beta a, \beta a]$, meaning they are indeed $(\ell_0 + \ell_\infty)$ bounded. Formally, we define these attacks as being bounded by an ℓ_0–norm of k, and an ℓ_∞–norm of βa, where β is a factor by which we scale the original domain $[-a, a]$. Since our goal is to develop a defense against a true ℓ_0–attack, unless otherwise stated, we set $\beta = 100$ as this effectively removes the ℓ_∞ constraint.

We compare with two defense methods: *Analysis by Synthesis* (ABS) model [28] and randomized ablation [30]. The ABS model relies on optimization-based inference by using variational auto-encoders that take 50 steps of gradient descent, repeating this 1000 times for each prediction. Randomized ablation use thousands of ablated samples for each input to construct a set of images, following which the classifier performs a majority vote on this set to decide the best label for the original image. Both of these methods are computationally costly, while our method's complexity comes from the first k–truncated FC layer, where if the input array has dimension d, removing the top and bottom k only adds $O(d)$ (when k is constant) more operations per neuron, which is small compared to the overall complexity of deep neural networks.

For the ABS model on MNIST, using `sparse-rs` with an ℓ_0–budget of 12 and a time budget of $10,000$ the robust accuracy decreases to 45%, which was significantly lower than the previously reported 78%. Additionally, the `Pointwise Attack` was used to calculate ρ to be 22 pixels. Note that both of these results were achieved for $\beta = 1$, when testing these statistics for higher $\beta \in (1, 100]$ we found that the robust guarantees vanished within the first hundred iterations i.e., the robust accuracy became 0%, and ρ became 1 pixel. For methods utilizing randomized ablation, robust guarantees were improved in relation to the ABS model: ρ was reported to be 31 pixels when $\beta = 1$. Using code provided by the authors [30], we were able to confirm that $\beta = 1$ was used in their experiments, unfortunately we could not test their robust accuracy with the stronger `sparse-rs` framework, nor could we increase β to see if their defense would break similar to the ABS model. Due to these reasons, and the fact that truncation can act independently of ablation, we do not compare our results directly with theirs.

A. Results on MNIST and CIFAR

We first discuss our results when testing the proposed k–truncated FC network on the MNIST dataset. All networks $F^{(k)}$ were trained via stochastic gradient descent, and had the same architecture, consisting of 5 FC layers with ReLU activations between them, where the first layer was replaced with the k–truncated matrix transformation from (5). For adversarial training we used the `sparse-rs` attack with ℓ_0–budget of 10 pixels and time budget $t = 300$ queries. Using this attack on the training data itself, we derive a new set of adversarial examples which are all misclassified by our network $F^{(k)}$, and append this set to our training data. We then train on the appended dataset, repeatedly calculating a new set of adversarial examples every 25 epochs and appending them as well. Hence the adversarial examples are chosen according to a procedure which is adaptive w.r.t. to our network $F^{(k)}$, and we use this procedure as a means of solving the minimax problem in (2). We provide more details on the training framework used in the full version of the paper [41].

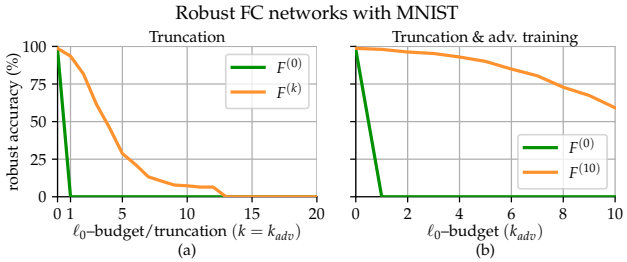

Fig. 1: In (a) we show the robust accuracy of our $F^{(k)}$ (orange) and $F^{(0)}$ (green) without adversarial training, where $k = k_{adv}$ is shown on the x-axis. We see that $F^{(k)}$ outperforms $F^{(0)}$, but at $k \geq 13$ the attack becomes too strong. In (b) we show the effect of adversarial training on $F^{(10)}$ (orange) and $F^{(0)}$ (green), varying the ℓ_0–budget on the x-axis as k. We can see as compared to without adversarial training, $F^{(10)}$ has substantially improved.

First, we look at the effect the truncation parameter k and ℓ_0–budget k_{adv} have on the initial robust accuracy, without adversarial training. We can see the strength of the attack portrayed in Fig. 1a, where the unprotected network $F^{(0)}$ fails for $k_{adv} \geq 1$, and even $F^{(k)}$ becomes fully susceptible to ℓ_0–attacks with budget $k_{adv} \geq 13$. We set out to improve the robustness of the specific classifier $F^{(10)}$ via adversarial training (using an ℓ_0–budget of 10 pixels), where we demonstrate this robustness by testing $F^{(10)}$ against ℓ_0–attacks with varying budgets $k_{adv} \leq 10$. Note that during adversarial training we use a time budget of $t = 300$ queries, and hence we believe our robust accuracy should be tested with an attack of similar time budget. However, we use a much larger time budget of $t = 5000$ queries for the results displayed in Table I, while in Fig. 1 we use $t = 1000$ queries.

We can see from Fig. 1b that adversarial training improves the robust accuracy of our k–truncated classifier, agreeing with our theory. When comparing to the initial results in Fig. 1a, adversarial training shows no effect on the robust accuracy of the regular classifier $F^{(0)}$, while displaying substantial improvements when applied to $F^{(10)}$.

We highlight these results in Table I, showing that for lower budgets k_{adv} we can maintain high robust accuracy even as the time budget t increases. Also, there is no loss in classification accuracy from truncation as both $F^{(0)}$ and $F^{(10)}$ reach the same clean accuracy after adversarial training, which is slightly lower than the base classifier's clean accuracy of 99.3%. Here we refer to the accuracy on the test set without adversarial examples as the *clean accuracy*, and the classifier derived when trained without an adversary as the *base classifier*. We note that for higher k_{adv} one can only expect so much improvement until the ℓ_0–attack becomes too powerful for any classifier, although we suspect tuning k and running the attack for longer while training can help improve robustness further.

To underline our results we refer to the `Pointwise Attack`, where we display in Table II the values of ρ for our classifiers. We ran 10 iterations of the attack, utilizing the entire test set of MNIST images. We confirm that $F^{(10)}$ outperforms its unprotected counterpart $F^{(0)}$, and does just as well as the ABS model even when $\beta = 1$ [28]. Since we know that both the ABS model and $F^{(0)}$ have no robustness guarantees when $\beta = 100$, we think it is significant that under this setting $F^{(10)}$ still achieves a high ρ of 17 pixels.

Setup			Rob. acc. `sparse-rs` (%)		
Net	Acc. (%)	ℓ_0–budget	$t = 3e2$	$t = 1e3$	$t = 5e3$
$F^{(0)}$	98.02	3, 5, 8	0.00	0.00	0.00
$F^{(10)}$	98.73	3	95.51	94.73	92.97
$F^{(10)}$	98.73	5	93.55	89.65	81.84
$F^{(10)}$	98.73	8	85.94	73.24	58.79
VGG$^{(0)}$	87.68	3	64.45	52.73	39.65
VGG$^{(0)}$	87.68	8	52.92	40.23	26.36
VGG$^{(10)}$	87.27	3	77.73	71.67	67.77
VGG$^{(10)}$	87.27	8	70.70	61.33	53.13

TABLE I: Adversarial training using `sparse-rs` on MNIST and CIFAR. The table above shows the final robust accuracy of $F^{(10)}$ and $F^{(0)}$ after adversarial training on MNIST, as well as VGG$^{(0)}$ VGG$^{(10)}$ on CIFAR. We give the clean accuracy (Acc. %) of the classifiers along with the ℓ_0–budget used to attack them. We then show the robust accuracy (Rob. acc.) as we vary the adversary's time budget t. Note $F^{(0)}$ fails for any budget greater than zero.

Setup		Median (pixels)	
Architecture	Dataset	$\beta = 100$	$\beta = 1$
$F^{(0)}$	MNIST	1	13
$F^{(10)}$	MNIST	**17**	**21**
VGG	CIFAR	2	3
VGG$^{(10)}$	CIFAR	11	17

TABLE II: ρ using the `Pointwise Attack`. The table above shows the median adversarial attack magnitude denoted as ρ for both our fully connected and convolution networks. Note that the ABS model as well as randomized ablation are not effective when $\beta = 100$, while for $\beta = 1$ the ABS model achieves an identical $\rho = 21$.

We believe our results for MNIST convey the efficiency and potential of utilizing truncation when designing robust classifiers. We also understand that in order to expand the applicability of truncation, we need to consider how it can be utilized within convolutional neural networks. Unlike with FC layers, the extension of truncation to $2d$-convolutional layers is heuristically motivated, where our approach is directly applying truncation before the first layer of VGG–19 [42].

As with FC networks, VGG$^{(0)}$ and its k–truncated counterpart VGG$^{(10)}$ were trained with an ℓ_0–budget $k_{adv} = 10$, and attacked with varying time budgets and ℓ_0–budgets. The results are displayed in Table I. We see that although VGG$^{(0)}$ is able to maintain a robust accuracy above 0% thanks to adversarial training, we can improve this by adding our truncation component. We also see that the clean accuracy did not suffer when utilizing truncation, and the end result was comparable to the base classifier's accuracy of approximately 91%. We think this is significant since prior methods showed large trade-offs between robust accuracy and test set performance [28], [30], while truncation combined with adversarial training does strictly better than adversarial training alone.

REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc.

[2] Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. Globally normalized transition-based neural networks. *arXiv preprint arXiv:1603.06042*, 2016.

[3] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[4] David Silver, Aja Huang, Christopher Maddison, Arthur Guez, Laurent Sifre, George Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489, 01 2016.

[5] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. *Lecture Notes in Computer Science*, page 387–402, 2013.

[6] 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.

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

[8] 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.

[9] 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.

[10] Francesco Croce and Matthias Hein. Sparse and imperceivable adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4724–4732, 2019.

[11] 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.

[12] Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world, 2016.

[13] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, pages 274–283. PMLR, 2018.

[14] Zhinus Marzi, Soorya Gopalakrishnan, Upamanyu Madhow, and Ramtin Pedarsani. Sparsity-based defense against adversarial attacks on linear classifiers. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 31–35. IEEE, 2018.

[15] Yang Bai, Yuyuan Zeng, Yong Jiang, Shu-Tao Xia, Xingjun Ma, and Yisen Wang. Improving adversarial robustness via channel-wise activation suppressing. In *International Conference on Learning Representations*, 2020.

[16] 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.

[17] Jérôme Rony, Luiz G Hafemann, Luiz S Oliveira, Ismail Ben Ayed, Robert Sabourin, and Eric Granger. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4322–4330, 2019.

[18] Chang Xiao, Peilin Zhong, and Changxi Zheng. Enhancing adversarial defense by k-winners-take-all. In *8th International Conference on Learning Representations*, 2020.

[19] J Lin, C Gan, and S Han. Defensive quantization: When efficiency meets robustness. *Artificial Intelligence, Communication, Imaging, Navigation, Sensing Systems*, page 8, 2019.

[20] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: elastic-net attacks to deep neural networks via adversarial examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[21] Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Sparsefool: a few pixels make a big difference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9087–9096, 2019.

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

[23] Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 631–648, 2018.

[24] Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John C. Duchi, and Percy Liang. Adversarial training can hurt generalization, 2019.

[25] 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.

[26] Adel Javanmard, Mahdi Soltanolkotabi, and Hamed Hassani. Precise tradeoffs in adversarial training for linear regression. In *Conference on Learning Theory*, pages 2034–2078. PMLR, 2020.

[27] 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.

[28] Lukas Schott, Jonas Rauber, Matthias Bethge, and Wieland Brendel. Towards the first adversarially robust neural network model on mnist. *arXiv preprint arXiv:1805.09190*, 2018.

[29] Francesco Croce, Maksym Andriushchenko, Naman D Singh, Nicolas Flammarion, and Matthias Hein. Sparse-rs: a versatile framework for query-efficient sparse black-box adversarial attacks. *arXiv preprint arXiv:2006.12834*, 2020.

[30] Alexander Levine and Soheil Feizi. Robustness certificates for sparse adversarial attacks by randomized ablation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4585–4593, 2020.

[31] Juncheng Li, Frank Schmidt, and Zico Kolter. Adversarial camera stickers: A physical camera-based attack on deep learning systems. In *International Conference on Machine Learning*, pages 3896–3904. PMLR, 2019.

[32] Kathrin Grosse, Nicolas Papernot, Praveen Manoharan, Michael Backes, and Patrick McDaniel. Adversarial perturbations against deep neural networks for malware classification. *arXiv preprint arXiv:1606.04435*, 2016.

[33] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? natural language attack on text classification and entailment. *arXiv preprint arXiv:1907.11932*, 2, 2019.

[34] Adi Shamir, Itay Safran, Eyal Ronen, and Orr Dunkelman. A simple explanation for the existence of adversarial examples with small hamming distance. *arXiv preprint arXiv:1901.10861*, 2019.

[35] Nina Narodytska and Shiva Prasad Kasiviswanathan. Simple black-box adversarial perturbations for deep networks. *arXiv preprint arXiv:1612.06299*, 2016.

[36] Payam Delgosha, Hamed Hassani, and Ramtin Pedarsani. Robust classification under ℓ_0 attack for the gaussian mixture model. *arXiv preprint arXiv:2104.02189, to appear in SIAM Journal on Mathematics of Data Science*, 2022.

[37] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[38] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

[39] Peter J Huber. *Robust statistics*, volume 523. John Wiley & Sons, 2004.

[40] Andrew L. Maas. Rectifier nonlinearities improve neural network acoustic models. 2013.

[41] Mark Beliaev, Payam Delgosha, Hamed Hassani, and Ramtin Pedarsani. Efficient and robust classification for sparse attacks, 2022.

[42] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.