frontiers in Artificial Intelligence

ORIGINAL RESEARCH
published: 04 January 2022
doi: 10.3389/frai.2021.780843

Adversarially Robust Learning *via* Entropic Regularization

Gauri Jagatap, Ameya Joshi, Animesh Basak Chowdhury, Siddharth Garg and Chinmay Hegde*

Electrical and Computer Engineering, New York University, New York, NY, United States

In this paper we propose a new family of algorithms, ATENT, for training adversarially robust deep neural networks. We formulate a new loss function that is equipped with an additional entropic regularization. Our loss function considers the contribution of adversarial samples that are drawn from a specially designed distribution in the data space that assigns high probability to points with high loss and in the immediate neighborhood of training samples. Our proposed algorithms optimize this loss to seek adversarially robust valleys of the loss landscape. Our approach achieves competitive (or better) performance in terms of robust classification accuracy as compared to several state-of-the-art robust learning approaches on benchmark datasets such as MNIST and CIFAR-10.

Keywords: adversarial learning, robustness, adversarial attack, regularization, neural network training

OPEN ACCESS

Edited by:
Yang Zhou,
Auburn University, United States

Reviewed by:
Ji Liu,
Baidu, China
Xingjian Li,
Baidu, China

****Correspondence:***
Gauri Jagatap
gauri.jagatap@nyu.edu

Specialty section:
This article was submitted to Machine Learning and Artificial Intelligence, a section of the journal Frontiers in Artificial Intelligence

Received: *21 September 2021*
Accepted: *28 October 2021*
Published: *04 January 2022*

Citation:
Jagatap G, Joshi A, Chowdhury AB, Garg S and Hegde C (2022) Adversarially Robust Learning via *Entropic Regularization. Front. Artif. Intell. 4:780843. doi: 10.3389/frai.2021.780843*

1 INTRODUCTION

Deep neural networks have led to significant breakthroughs in the fields of computer vision (Krizhevsky et al., 2012), natural language processing (Zhang et al., 2020), speech processing (Carlini et al., 2016), recommendation systems (Tang et al., 2019) and forensic imaging (Rota et al. (2016)). However, deep networks have also been shown to be very susceptible to carefully designed "attacks" (Goodfellow et al., 2014; Papernot et al., 2016; Biggio and Roli, 2018). In particular, the outputs of networks trained via traditional approaches are rather brittle to maliciously crafted perturbations in both input data as well as network weights (Biggio et al., 2013).

Formally put, suppose the forward map between the inputs x and outputs y is modeled *via* a neural network as $y = f(w; x)$ where w represents the set of trainable weight parameters. For a classification task, given a labeled dataset $\{x_i, y_i\}$, $i = 1, \ldots, n$ where X and Y represents all training data pairs, the standard procedure for training neural networks is to seek the weight parameters w that minimize the empirical risk:

$$\hat{w} = \arg\min_{w} \frac{1}{n} \sum_{i=1}^{n} L\left(f(w; x_i), y_i\right) := \arg\min_{w} \mathcal{L}(X; Y, w).$$

However, the prediction $\hat{y}(x) = f(\hat{w}; x)$ can be very sensitive to changes in both $\hat{w}$ and x. For example, if a bounded perturbation to a test image input (or to the neural network weights) is permitted, i.e., $\hat{y}_i = f(\hat{w}; x_i + \delta_i)$ where δ_i represents the perturbation, then the predicted label $\hat{y}_i$ can be made *arbitrarily* different from the true label y_i.

Several techniques for finding such adversarial perturbations have been put forth. Typically, this can be achieved by maximizing the loss function within a neighborhood around the test point x (Tramèr et al., 2017; Madry et al., 2018):

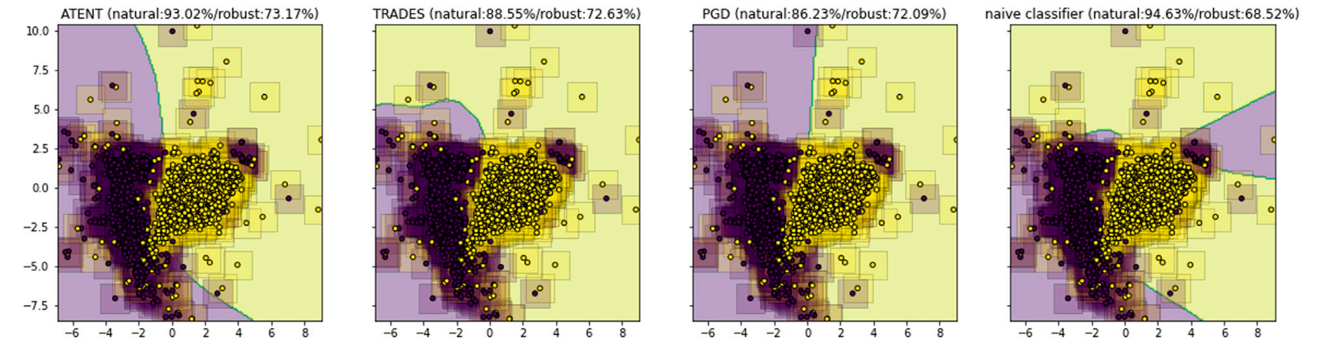

FIGURE 1 | TSNE visualization of decision boundaries for a 3-layer neural network trained using different defenses; corresponding natural and robust test accuracies against ℓ_∞ attacks for classifying MNIST digits 5 and 8.

$$\bar{x}_{\text{worst}} = \arg\max_{\delta\in\Delta_p} L\left(f\left(\hat{w}; x+\delta\right), y\right), \tag{1}$$

where $\hat{w}$ are the final weights of a pre-trained network. The perturbation set Δ_p is typically chosen to be an ℓ_p-ball for some $p \in \{0, 1, 2, \infty\}$.

The existence of adversarial attacks motivates the need for a "defense" mechanism that makes the network under consideration more robust. Despite a wealth of proposed defense techniques, the jury is still out on how optimal defenses should be constructed (Athalye et al., 2018).

We discuss several families of effective defenses. The first involves *adversarial training* (Madry et al., 2018). Here, a set of adversarial perturbations of the training data is constructed by solving a min-max objective of the form:

$$\hat{w} = \min_{w} \max_{\delta\in\Delta_p} \frac{1}{n}\sum_{i=1}^{n} L\left(f\left(w; x_i+\delta\right), y_i\right).$$

Wong and Kolter (2018) use a convex outer adversarial polytope as an upper bound for worst-case loss in robust training; here the network is trained by generating adversarial as well as few non-adversarial examples in the convex polytope of the attack via a linear program. Along the same vein include a mixed-integer programming based certified training for piece-wise linear neural networks (Tjeng et al., 2018) and integer bound propagation (Gowal et al., 2019).

The last family of approaches involves *randomized smoothing*. Here, both training the network as well as the inference made by the network are smoothed out over several stochastic perturbations of the target example (Lecuyer et al., 2019; Cohen et al., 2019; Salman et al., 2019a). This has the effect of optimizing a smoothed-adversarial version of the empirical risk. Randomized smoothing has also been used in combination with adversarial training (Salman et al., 2019b) for improved adversarial robustness under ℓ_2 attacks[1].

In this paper, we propose a new approach for training adversarially robust neural networks. The key conceptual ingredient underlying our approach is *entropic regularization*. Borrowing intuition from Chaudhari et al. (2019), instead of the empirical risk (or its adversarial counterpart), our algorithm instead optimizes over a local entropy-regularized version of the empirical risk:

$$\begin{aligned} \hat{w} &= \arg\min_{w} \mathcal{L}_{DE}, \\ \mathcal{L}_{DE} &= \int_{X'} \mathcal{L}\left(X'; Y, w\right) \left[\frac{e^{\left(\mathcal{L}\left(X'; Y, w\right) - \frac{\gamma}{2}\|X - X'\|_p^p\right)}}{Z}\right] dX'. \end{aligned} \tag{2}$$

Intuitively, this new loss function can be viewed as the convolution of the empirical risk with a Gibbs-like distribution to sample points from the neighborhoods, X', of the training data points X that have high loss. Therefore, compared to adversarial training, we have replaced the inner maximization with an expected value with respect to a modified Gibbs measure which is matched to the geometry of the perturbation set.

Since the above loss function is difficult to optimize (or even evaluate exactly), we instead approximate it via Monte Carlo techniques. In particular, we use Stochastic Gradient Langevin Dynamics (Welling and Teh, 2011); in this manner, our approach blends in elements from adversarial training, randomized smoothing, and entropic regularization. We posit that the combination of these techniques will encourage a classifier to learn a better robust decision boundary as compared to prior art (see visualization in **Figure 1**).

To summarize, our specific contributions are as follows:

1. We propose a new entropy-regularized loss function for training deep neural networks (**Eq. 2**) that is a robust version of the empirical risk.
2. We propose a new Monte Carlo algorithm to optimize this new loss function that is based on Stochastic Gradient Langevin Dynamics. We call this approach Adversarial Training with ENTropy (ATENT).

[1] This family of methods has the additional benefit of being certifiably robust: all points within a ball of a given radius around the test point are provably classified with the correct label.

3. We show that ATENT-trained networks provide improved (robust) test accuracy when compared to existing defense approaches.
4. We combine randomized smoothing with ATENT to show competitive performance with the smoothed version of TRADES.

In particular, we are able to train an ℓ_∞-robust CIFAR-10 model to 57.23% accuracy at PGD attack level $\epsilon = 8/255$, which is higher than the latest benchmark defenses based on both adversarial training using early stopping (Salman et al., 2019b) (56.8%) as well as TRADES (56.6%) (Zhang et al., 2019b).

2 PRIOR WORK

Evidence for the existence of adversarial inputs for deep neural networks is by now well established (Carlini N. and Wagner D. A., 2017; Dathathri et al., 2017; Goodfellow et al., 2015; Goodfellow, 2018; Szegedy et al., 2013; Moosavi-Dezfooli et al., 2017). In image classification, the majority of attacks have focused on the setting where the adversary confounds the classifier by adding an imperceptible perturbation to a given input image. The range of the perturbation is pre-specified in terms of bounded pixel-space ℓ_p-norm balls. Specifically, an ℓ_p- attack model allows the adversary to search over the set of input perturbations $\Delta_{p,\epsilon} = \{\delta\colon \|\delta\|_p \leq \epsilon\}$ for $p = \{0, 1, 2, \infty\}$.

Initial attack methods, including the Fast Gradient Sign Method (FGSM) and its variants (Goodfellow et al., 2014; Kurakin et al., 2016), proposed techniques for generating adversarial examples by ascending along the sign of the loss gradient:

$$x_{adv} = x + \epsilon \text{sgn}\left(\nabla_x L\left(f\left(\hat{w}; x\right), y\right)\right),$$

where $(x_{adv} - x) \in \Delta_{\infty,\epsilon}$. Madry et al. (2018) proposed a stronger adversarial attack via projected gradient descent (PGD) by iterating FGSM several times, such that

$$x^{t+1} = \Pi_{x+\Delta_{p,\epsilon}}(x^t + \alpha \text{sgn}\left(\nabla_x L\left(f\left(\hat{w}; x\right), y\right)\right),$$

where $p = \{2, \infty\}$. These attacks are (arguably) the most successful available attack techniques reported to date, and serve as the starting point for our comparisons. Both Deep Fool Moosavi-Dezfooli et al., 2016) and Carlini-Wagner (Carlini N. and Wagner D., 2017) construct an attack by finding smallest possible perturbation that can flip the label of the network output.

Several strategies for defending against attacks have been developed. In Madry et al. (2018), adversarial training is performed via the min-max formulation **Eq. 1**. The inner maximization is solved using PGD, while the outer objective is minimized using stochastic gradient descent (SGD) with respect to w. This can be slow to implement, and speed-ups have been proposed in Shafahi et al. (2019) and Wong et al. (2020). In Li B. et al. (2018); Cohen et al. (2019); Lecuyer et al. (2019); Salman et al. (2019a, Salman et al. (2019b), the authors developed certified defense strategies via randomized smoothing. This approach consists of two stages: the first stage consists of training with noisy samples, and the second stage produces an ensemble-based inference. See Ren et al. (2020) for a more thorough review of the literature on various attack and defense models.

Apart from minimizing the worst case loss, approaches which minimize the upper bound on worst case loss inclu Wong et al., 2018; Tjeng et al. (2018); Gowal et al. (2019). Another breed of approaches use a modified loss function which considers surrogate adversarial loss as an added regularization, where the surrogate is cross entropy (Zhang et al., 2019b) (TRADES), maximum margin cross entropy (Ding et al., 2019) (MMA) and KL divergence (Wang et al., 2019) (MART) between adversarial sample predictions and natural sample predictions.

In a different line of work, there have been efforts towards building neural network networks with improved generalization properties. In particular, heuristic experiments by Hochreiter and Schmidhuber (1997); Keskar et al. (2016); Li H. et al. (2018) suggest that the loss surface at the final learned weights for well-generalizing models is relatively "flat"[2]. Building on this intuition, Chaudhari et al. (2019) showed that by explicitly introducing a smoothing term (via entropic regularization) to the training objective, the learning procedure weights towards regions with flatter minima by design. Their approach, Entropy-SGD (or ESGD), is shown to induce better *generalization* properties in deep networks. We leverage this intuition, but develop a new algorithm for training deep networks with better *adversarial robustness* properties. We also highlight some papers written concurrently in **Supplementary Material**.

3 PROBLEM FORMULATION

The task of classification, given a training labelled dataset $\{x_i \in \mathcal{X}, y_i\}$, $i \in \{1, \ldots, n\}$, consists of solving the standard objective by optimizing weight parameters w, $\min_w \frac{1}{n}\sum_{i=1}^n L(f(w; x_i), y_i)$ where y_i is a one-hot class encoding vector of length m and m is the total number of classes. The training data matrix itself is represented using shorthand $X \in \mathbb{R}^{n \times d}$ and labels in $Y \in \mathbb{R}^n$ where we have access to n training samples which are d-dimensional each. Given this formulation, the primary task is to minimize the cross-entropy Loss function $\mathcal{L}(w; X, Y) = -\frac{1}{n}\sum_{i=1}^n \sum_{j=1}^m y_{i,j} \log \hat{y}_{i,j}$. In this paper, we design an augmented version of the loss function $\mathcal{L}$ which models a class of adversarial perturbations and also introduce a new procedure to minimize it.

We first recap the Entropy SGD (Chaudhari et al., 2019) (see also **Supplementary Material**). Entropy-SGD considers an augmented loss function of the form

[2] This is not strictly necessary, as demonstrated by good generalization at certain sharp minima (Dinh et al., 2017).

$$\mathcal{L}_{ent}(w;X,Y) = -\log\int_{w'} e^{-\mathcal{L}\left(w';X,Y\right)-\frac{\gamma}{2}\|w-w'\|_2^2}dw'.$$

By design, minimization of this augmented loss function promotes minima with wide valleys. Such a minimum would be robust to perturbations in w, but is not necessarily advantageous against adversarial data samples x_{adv}. In our experiments (**Section 4**) we show that networks trained with Entropy-SGD perform only marginally better against adversarial attacks as compared to those trained with standard SGD.

For the task of adversarial robustness, we instead develop a data-space version of Entropy-SGD. To model for perturbations in the samples, we design an augmented loss that regularizes the data space. Note that we only seek specific perturbations of data x that *increase* the overall loss value of prediction. In order to formally motivate our approach, we first make some assumptions.

Assumption 1. The distribution of possible adversarial data inputs of the neural network obeys a positive exponential distribution of the form below, where the domain of $\mathcal{L}(X;Y,w)$ is bounded:

$$p(X;Y,w,\beta) = \begin{cases} Z_{w,\beta}^{-1}e^{\beta\mathcal{L}(X;Y,w)} & \text{if} \quad \mathcal{L}(X;Y,w) \le R, \\ 0 & \text{if} \quad \mathcal{L}(X;Y,w) > R, \end{cases} \quad (3)$$

and Zw,β is the partition function that normalizes the probability distribution.

Note here that cross entropy loss $\mathcal{L}$ is always lower bounded as $\mathcal{L} \ge 0$. flushleft

Intuitively, the neural network is more likely to "see" perturbed examples from the adversary corresponding to higher loss values as compared to lower loss values. The parameter R is chosen to ensure that the integral of the probability curve is bounded. When the temperature parameter $\beta \rightarrow \infty$, the above Gibbs distribution concentrates at the maximizer(s) of $\mathcal{L}(\bar{X};Y,w)$, where $\bar{X}$ is the "worst possible" set of adversarial inputs to the domain of the loss function for fixed weights w. For a given attack ball $\Delta_{p,}\epsilon$ with radius ϵ and norm p, and fixed weights w, this value equates to:

$$\bar{X} = \arg\max_{X' \in \Delta_{p,\epsilon}} \mathcal{L}\left(X';X,Y,w\right),$$

where $\max_{X' \in \Delta_{p,\epsilon}} \mathcal{L}(X';X,Y,w) \le R$.

Assumption 2. A modified distribution, (without loss of generality, setting $\beta = 1$) with an additional smoothing parameter, assumes the form:

$$p\left(X';X,Y,w,\gamma\right) = \begin{cases} Z_{X,w,\gamma}^{-1}e^{\mathcal{L}(X';Y,w)-\frac{\gamma}{2}\|X'-X\|_F^2} & \text{if} \quad \mathcal{L}\left(X';Y,w\right) \le R \\ 0 & \text{if} \quad \mathcal{L}\left(X';Y,w\right) > R \end{cases}$$

where $Z_{X,w,\gamma}$ is the partition function that normalizes the probability distribution.

Here γ controls the penalty of the distance of the adversary from true data X; if $\gamma \rightarrow \infty$, the sampling is sharp, i.e. $p(X' = X; X, Y, w, \gamma) = 1$ and $p(X' \neq X; X, Y, w, \gamma) = 0$, which is the same as sampling only the standard loss $\mathcal{L}$, meanwhile $\gamma \rightarrow 0$ corresponds to a uniform contribution from all possible data points in the loss manifold.

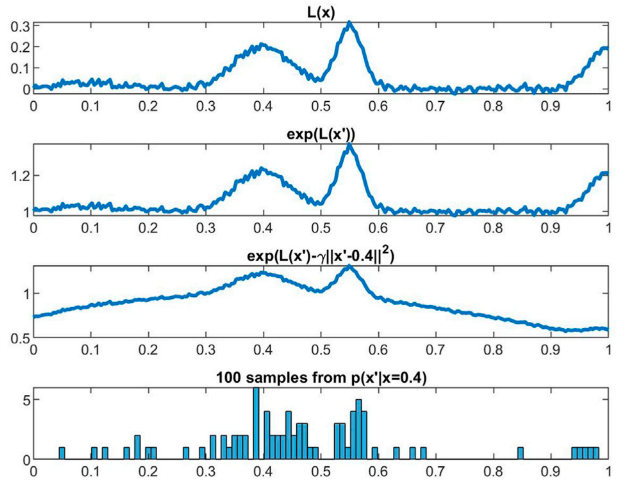

FIGURE 2 | Illustration of the sampling procedure in Assumption 2 at fixed weights w. The distribution produces samples x' from distribution $p(x'|x = 0.4)$, and we compute the average loss over these samples. Effectively, this encourages ATENT to search for w where $L(x; w)$ is relatively flat in the neighborhood of x.

Now, we develop an augmented loss function which incorporates the probabilistic formulation in Assumption 2. The standard objective can be re-written as the functional convolution:

$$\min_w \mathcal{L}(w;X,Y) := \min_w \int_{X'} \mathcal{L}\left(X';Y,w\right)\delta\left(X-X'\right)dX',$$

which can be seen as a sharp sampling of the loss function at training points X. Now, define the *Data-Entropy Loss*:

$$\mathcal{L}_{DE}(w;X,Y,\gamma) = \int_{X'} \mathcal{L}\left(X';Y,w\right)p\left(X';X,Y,w,\gamma\right)dX' \quad (4)$$

our new objective is to minimize this augmented objective function $\mathcal{L}_{DE}(w;X,Y,\gamma)$, which resembles expected value of the standard loss function sampled according to a distribution that (i) penalizes points further away from the true training data 2) boosts data points which correspond to high loss values. Specifically, the adversarial samples generated by the distribution in Assumption 2 will correspond to those with *high loss* values in the *immediate neighborhood* of the true data samples. This sampling process is also described in **Figure 2**. We also highlight theoretical properties of our augmented loss function *via* Lemma 3.1, proof of which can be found in of supplement B.

Lemma 3.1. The effective loss $F(X';X,Y,w) := \frac{\gamma}{2}\|X-X'\|_F^2 - \mathcal{L}(X';Y,w)$ which guides the Langevin sampling process in **Eq. 8** is

1. $\beta + \gamma$ smooth if $\mathcal{L}(X; Y, w)$ is β-smooth in X.
2. $(\frac{\gamma}{4}, \frac{L^2}{\gamma} + \frac{\gamma}{2}\|X\|_F^2)$ dissipative if $\mathcal{L}(X; Y, w)$ is L-Lipschitz in X.

If gradient descent is used to minimize the loss in **Eq. 4**, the gradient update corresponding to the augmented loss function can be computed as follows

$$\begin{aligned}\nabla_w \mathcal{L}_{de}(w; X, Y, \gamma) &= \nabla_w \int_{X'} \mathcal{L}\left(X'; Y, w\right) p\left(X'; X, Y, w, \gamma\right) dX' \\ &= \nabla_w \mathbb{E}_{X' \sim p\left(X'; X, Y, w, \gamma\right)} \left[\mathcal{L}\left(X'; Y, w\right)\right]\end{aligned} \tag{5}$$

Correspondingly, the weights of the network, when trained using gradient descent, using **Eq. 5**, can be updated as

$$w^+ = w - \eta \nabla_w \mathbb{E}_{X' \sim p\left(X'; X, Y, w, \gamma\right)} \left[\mathcal{L}\left(X'; Y, w\right)\right] \tag{6}$$

where η is the step size. The expectation in **Eq. 5** is carried out over the probability distribution of data samples X' as defined in Assumption 2. This can be seen as the adversarial version of the formulation developed in Entropy SGD (Chaudhari et al. (2019)), where the authors use a Gibbs distribution to model an augmented loss function that explores the loss surface at points that are perturbed from the current weights w, denoted by w' (see **Supplementary Material**). In contrast, in our approach, we consider loss contributions from perturbations X' of data points X. This analogue is driven by the fact that the core objective in Chaudhari et al. (2019) is to design a network which is robust to perturbations in *weights* (generalization), where as the core objective of this paper is to design a network that is robust to perturbations in the *inputs* (adversarial robustness).

The expectation in **Eq. 5** is computationally intractable to optimize (or evaluate). However, using the Euler discretization of the Langevin Stochastic Differential Equation (Welling and Teh, 2011), it can be approximated well. Samples can be generated from $p(X')$ as:

$$X'^{k+1} = X'^{k} + \eta' \nabla_{X'} \log p\left(X'^{t}\right) + \sqrt{2\eta'}\varepsilon \mathcal{N}(0, \mathbb{I}) \tag{7}$$

where η' is the step size for Langevin sampling, ε is a scaling factor that controls additive noise. In Langevin dynamics, when one considers a starting point of X'^0 then the procedure above yields samples $X'^1 \ldots X'^t$ that follow the distribution $p(X')$. Intuitively, the stochastic process X'^t is more likely to visit points in the immediate neighborhoods of the *entire training* dataset X corresponding to high loss values.

Observe that X' and X have the same dimensions and the gradient term in the above equation needs to be computed over n, d-dimensional data points. In practice this can be computationally expensive. Therefore, we discuss a stochastic variant of this update rule, which considers mini-batches of training data instead. Plugging in the distribution in **Eq.3**, and using the Euler discretization for Langevin Stochastic Differential Equations, the update rule for sampling X' is

$$\begin{aligned}\frac{X'^{k+1} - X'^{k}}{\eta'} &= \nabla_{X'^k}\left(\mathcal{L}\left(X'^k; Y, w\right) - \frac{\gamma}{2}\|X - X'^k\|_F^2\right) + \sqrt{\frac{2\varepsilon^2}{\eta'}}\mathcal{N}(0, \mathbb{I}) \\ &= \nabla_{X'^k}\mathcal{L}\left(X'^k; Y, w\right) + \gamma\left(X - X'^k\right) + \sqrt{\frac{2\varepsilon^2}{\eta'}}\mathcal{N}(0, \mathbb{I})\end{aligned} \tag{8}$$

where we have incorporated $Z_{X,w\gamma}$ in the step size η'. Note that as the number of updates $k \rightarrow \infty$, the estimates from the procedure in **Eq. 8** converge to samples from the true distribution. $p(X'; X, Y, w, \gamma)$. We then want to estimate $\nabla_w \mathcal{L}_{de}(w; X, Y, \gamma) = \nabla_w \mathbb{E}_{X' \sim p(X')}[\mathcal{L}(w; X', Y, \gamma)]$ using the samples obtained from the above iterative procedure. Chaudhari et al. (2019), use an exponentially decaying averaging process to estimate the expected value.

Batch-wise updates for stochastic gradient estimates: As is typical with large datasets, instead of using the entire training data for computing gradients in **Eq. 5** and **Eq. 7**, one can use batch-wise data where the training data is segmented into J batches $[X_{B_1}, X_{B_2} \ldots X_{B_J}]$. This is essentially a combination of Stochastic Gradient Descent and Langevin Dynamics and is known as Stochastic Gradient Langevin Dynamics in recent literature Welling and Teh, 2011).

This discussion effectively leads to the algorithm shown in **Algorithm 1**, which we refer to as Adversarial Training using Entropy (or ATENT), designed for ℓ_2 attacks. Note that we have considered exponentially decaying averaging over sample loss μ^k in Line 10 of **Algorithm 1**.

Algorithm 1. ℓ_2-ATENT

```
1:  Input: $X = [X_{B_1}, X_{B_2} \ldots X_{B_J}], f, \eta, \eta', w = w^0, \gamma, \varepsilon, \alpha$
2:  for $t = 1, \cdots T$ do
      (outer loop of SGD)
3:    for $j = 1, \cdots J$ do
        (scan through all batches of data)
4:      $x_i^0 \leftarrow x_i + \delta_i$ {$\forall x_i \in X_{B_j}$, $K$ is number of samples generated using Langevin dynamics}
5:      $\mu^j \leftarrow 0$
6:      for $k = 1, \cdots, K$ do
7:        $dx'^k \leftarrow \frac{1}{n_j}\sum_{i=1}^{n_j} \nabla_{x=x'^k} L(f(w^t; x)) + \gamma(x^k - x'^k)$
8:        $x'^{k+1} \leftarrow x'^k + \eta' dx'^k + \sqrt{2\eta'}\varepsilon\mathcal{N}(0,1)$ {Langevin update}
9:        $\mu^k \leftarrow \frac{1}{B}\sum_{x_i \in X_{B_j}} L(w^t; x'^{k+1})$ {augmented batch loss for $X_{B_j}$}
10:       $\mu^j \leftarrow (1-\alpha)\mu^j + \alpha\mu^k$
11:     end for
12:     $dL^t \leftarrow \nabla_w \mu^j$
13:     $w^{t+1} \leftarrow w^t - \eta dL^t$
14:   end for
15: end for
16: Output $\hat{w} \leftarrow w^T$
```

Comparison to PGD Adversarial Training: (see also Algorithm 2 Madry et al., 2018 in **Supplementary Material**, referred as PGD-AT). It is easy to see that the updates of PGD-AT are similar to that of **Algorithm 1**, consisting broadly of two types of gradient operations in an alternating fashion—1) an (inner) gradient with respect to samples X (or batch-wise samples X_{B_j}) and 2) an (outer) gradient with respect to weights w. While PGD-AT minimizes the *worst-case* loss in an ϵ-neighborhood (specifically ℓ_2 or ℓ_∞ ball) of X, ATENT minimizes an *average loss* over our specifically designed probability distribution (Assumption 3) in the neighborhood of X. Note that the gradient operation in **Eq. 8** is also the

[1] This family of methods has the additional benefit of being certifiably robust: all points within a ball of a given radius around the test point are provably classified with the correct label.

gradient for the regularized version of inner maximation of the adversarial training problem (Madry et al., 2018), but with added noise term,

$$\max_{X'} \mathcal{L}\left(X'; X, Y, w\right) \ s.t.\ \|X' - X\|_F^2 \leq \epsilon \Leftrightarrow \max_{X'} \mathcal{L}\left(X'; X, Y, w\right) - \frac{\gamma}{2}\|X' - X\|_F^2 \quad (9)$$

constraint being satisfied if $\|X' - X\|_F$ is minimized, or $-\|X' - X\|_F$ is maximized).

The width of the Gaussian smoothing is adjusted with γ, which is analogous to controlling the projection radius ϵ in the inner-maximization of PGD-AT. Then the second and third terms in **Eq. 8** are simply gradient of an ℓ_2-regularization term over data space X' and noise. In this way, ATENT can be re-interpreted as a stochastic formalization of ℓ_2-PGD-AT, with noisy controlled updates.

Comparison to randomized smoothing: Cohen et al. (2019), describe a defense to adversarial perturbations, in the form of smoothing. A smoothed classifier g, under isotropic Gaussian noise $\varepsilon = \mathcal{N}(0, \sigma^2\mathbb{I})$, produces an output:

$$g(x) = \arg\max_j \mathbb{P}\left(f(x + \varepsilon) = j\right). \quad (10)$$

where $\mathbb{P}$ denotes probability distribution (see **Supplementary Material** for detailed discussion). SmoothAdv Salman et al., 2019b) is an adversarial attack as well as defense for smoothed classifiers, which replaces standard loss with cross entropy loss of a smoothed classier. In comparison, we compute a smoothed version of the cross entropy loss of a standard classifier. This is similar to the setup of Blum et al. (2020) (TRADES with smoothing). The procedure in **Algorithm 1** is therefore amenable to randomized smoothing in its evaluation. We discuss a smoothed evaluation of ATENT in the next section.

Algorithm 2. ℓ_∞-ATENT

```
Input: $X = [X_{B_1}, X_{B_2} \dots X_{B_J}], f, \eta, \eta', w = w^0, \gamma, \varepsilon, \alpha$
for $t = 1, \cdots T$ do
    (outer loop of SGD)
    for $j = 1, \cdots J$ do
        (scan through all batches of data)
        $x_i^0 \leftarrow x_i + \delta_i$ {$\forall x_i \in X_{B_j}, K$ is number of samples generated using Langevin dynamics}
        $\mu^j \leftarrow 0$
        for $k = 1, \cdots, K$ do
            $dx'^k \leftarrow \frac{1}{n_j}\sum_{i=1}^{n_j} \nabla_{x=x'^k} L(f(w^t; x))$
            $x'^{k+1} \leftarrow x'^k + P_\gamma^K(\eta' dx'^k + \sqrt{2\eta'}\varepsilon\mathcal{N}(0,1))$ {update follows Eq.12, projection active in $K^{th}$ iteration only.}
            $\mu^k \leftarrow \frac{1}{B}\sum_{x_i \in X_{B_j}} L(w^t; x'^{k+1})$ {augmented batch loss for $X_{B_j}$}
            $\mu^j \leftarrow (1-\alpha)\mu^j + \alpha\mu^k$
        end for
        $dL^t \leftarrow \nabla_w \mu^j$
        $w^{t+1} \leftarrow w^t - \eta dL^t$
    end for
end for
Output $\hat{w} \leftarrow w^T$
```

Extension to defense against ℓ_∞-attacks: It is evident that due to the isotropic structure of the Gibbs measure around each data point, **Algorithm 1**, ℓ_2-ATENT is best suited for ℓ_2 attacks. However this may not necessarily translate to robustness against ℓ_∞ attacks. For this case, one can use an alternate assumption on the distribution of potential adversarial examples. flushleft

Assumption 3. We consider a modification of the distribution in Assumption 2 to account for robustness against ℓ_∞ type attacks:

$$p\left(X'; X, Y, w, \gamma\right) = \begin{cases} Z_{X,w,\gamma}^{-1} e^{\left(\mathcal{L}\left(X'; Y, w\right) - \frac{\gamma}{2}\|X' - X\|_\infty\right)} & \text{if} \quad \mathcal{L}\left(X'; Y, w\right) \leq R \\ 0 & \text{if} \quad \mathcal{L}\left(X'; Y, w\right) > R \end{cases}$$

where $\|\cdot\|_\infty$ is the ℓ_∞ norm on the vectorization of its argument and $Z_{X,w,\gamma}$ normalizes the probability.

The corresponding Data Entropy Loss for ℓ_∞ defenses is: flushleft

$$\mathcal{L}_{DE,\infty}(w; X, Y) = Z_{X,w,\gamma}^{-1} \int_{X'} \mathcal{L}\left(X'; Y, w\right) e^{\left(\mathcal{L}\left(X'; Y, w\right) - \frac{\gamma}{2}\|X - X'\|_\infty\right)} dX'$$

This resembles a smoothed version of the loss function with a exponential ℓ_∞ kernel along the data dimension to model points in the ℓ_∞ neighborhood of X which have high loss. The SGD update to minimize this loss becomes:

$$\begin{aligned} \nabla_w \mathcal{L}_{DE,\infty}(w; X, Y) &= \nabla_w \mathbb{E}_{X' \sim p(X')}\left[\mathcal{L}\left(w; X', Y\right)\right] \\ &\Rightarrow w^+ = w - \eta\nabla_w \mathcal{L}_{DE,\infty}(w; X, Y) \end{aligned}$$

where the expectation over $p(X')$ is computed by using samples generated via Langevin Dynamics:

$$X'^{k+1} = X'^k + \eta'\nabla_{X'} \log p\left(X'^k\right) + \sqrt{2\eta'}\varepsilon\mathcal{N}(0, \mathbb{I})$$

Plugging in the distribution in Assumption 3 the update rule for sampling X':

$$\begin{aligned} \frac{X'^{k+1} - X'^k}{\eta'} &= \nabla_{X'^k}\left(L\left(X'^k; Y, w\right) - \frac{\gamma}{2}\|X - X'^k\|_\infty\right) + \sqrt{\frac{2\varepsilon^2}{\eta'}}\mathcal{N}(0, \mathbb{I}) \\ &= \nabla_{X'^k} L\left(X'^k; Y, w\right) + \gamma \text{sign}\left(X_i - X_i'^k\right) \cdot \mathbf{1} + \sqrt{\frac{2\varepsilon^2}{\eta'}}\mathcal{N}(0, \mathbb{I}) \end{aligned} \quad (11)$$

where $i = \text{argmax}_j|X_j - X_j'^k|$ and j scans all elements of the tensors X, X'^k and $\mathbf{1}_j = \delta_{i,j}$. The second term in the update rule navigates the updates X'^{k+1} to lie in the immediate ℓ_∞ neighborhood of X. Note that this training process requires taking gradients of ℓ_∞ distance. In the update rule in **Eq. 11**, the gradient update only happens along one coordinate. In practice when we test this update rule, the algorithm fails to converge. This is due to the fact that typically a sizeable number of elements of $X' - X$ have a large magnitude.

The expression in the penultimate step of **Eq. 11**, is the gradient of a regularized maximization problem,

$$\begin{aligned} &\max_{X'} \mathcal{L}(X'; X, Y, w) \\ &s.t. \quad \|X' - X\|_\infty \leq \epsilon \Leftrightarrow \max_{X'} \mathcal{L}(X'; X, Y, w) - \gamma\|X' - X\|_\infty \end{aligned}$$

where γ is inversely proportional to ϵ (constraint is satisfied if $\|X' - X\|_\infty$ is minimized, or $-\|X' - X\|_\infty$ is maximized). This expression can be maximized only if $X' \in \Delta_{\infty,\epsilon}$ of X; however when we take gradients along only one coordinate, this may not be sufficient to drive all coordinates of X' towards $\Delta_{\infty,\epsilon}$ of X.

Similar to the ℓ_∞ Carlini Wagner attack (Carlini N. and Wagner D., 2017), we replace the gradient update of the ℓ_∞ term, with a

TABLE 1 | Robust percentage accuracies of 5-layer convolutional net for MNIST against ℓ_2, $\epsilon = 2$ attack.

Attack → **↓ Defense**	**Benign Acc**	**ℓ_2 PGD-40**	**ℓ_2 CW**
SGD	**99.38**	19.40	13.20
Entropy SGD	99.24	19.12	14.52
ℓ_2 PGD-AT	98.76	72.94	-
TRADES	97.54	**76.08**	-
MMA	**99.27**	73.02	72.72
ℓ_2 ATENT	98.66	**77.21**	**76.72**

The highest two accuracy values in each column are highlighted in bold.

TABLE 2 | Robust accuracies (in percentages) of 5-layer convolutional net for MNIST against ℓ_∞, $\epsilon = 0.3$ attack.

Attack → **↓ Defense**	**Benign** **Acc**	**ℓ_∞ PGD-20** **$\epsilon_\infty = 0.3$**	**ℓ_∞ CW** **$\epsilon_\infty = 0.3$**
SGD	99.39	0.97	32.37
Entropy SGD	99.24	1.17	34.34
ℓ_∞ PGD-AT	99.36	96.01	94.25
TRADES	**99.48**	96.07	94.03
MMA	98.92	95.25	94.77
MART	98.74	**96.48**	**96.10**
ℓ_∞ ATENT	**99.45**	**96.44**	**97.40**

The highest two accuracy values in each column are highlighted in bold.

clipping based projection oracle. We design an accelerated version of the update rule in **Eq. 11**, in which we perform a clipping operation, i.e., an ℓ_∞ ball projection of the form:

$$
\begin{aligned}
X'^{k+1} - X'^{k} &= \eta' \nabla_{X'} L\left(X'^{k}; Y, w\right) + \sqrt{2\eta'}\varepsilon\mathcal{N}(0, \mathbb{I}),\\
X'^{K} - X'^{K-1} &= P\gamma\left(\eta' \nabla_{X'} L\left(X'^{K-1}; Y, w\right) + \sqrt{2\eta'}\,\varepsilon\mathcal{N}(0, \mathbb{I})\right)
\end{aligned}
\tag{12}
$$

where element-wise projection $P_\gamma(z) = z$ if $|z| < 1/\gamma$ and $P_\gamma(z) = 1/\gamma$ if $|z| > 1/\gamma$. Empirically, we also explored an alternate implementation where the projection takes place in each inner iteration k, however, we find the version in **Algorithm 2** to give better results.

In both **Algorithms 1** and **2**, we initialize the Langevin update step with a random normal perturbation δ_i of benign samples, which is constructed to lie inside within approximately $1/\gamma$ radius of the natural samples.

4 EXPERIMENTS

In this section we perform experiments on a five-layer convolutional model with 3 CNN and 2 fully connected layers, used in Zhang et al. (2019b); Carlini N. and Wagner D. (2017), trained on MNIST. We also train a WideResNet-34-10 on CIFAR10 [as used in Zhang et al. (2019b)] as well as ResNet20. Due to space constraints, we present supplemental results in **Supplementary Material**. We conduct our experiments separately on networks specifically trained for ℓ_2 attacks and those trained for ℓ_∞ attacks. We also test randomized smoothing for our ℓ_2-ATENT model. Source code is provided in the supplementary material.

TABLE 3 | Robust accuracies of WRN34-10 net for CIFAR10 against ℓ_∞ attack of $\epsilon = 8/255$.

Defense → **↓ Attack**	**PGD** **AT**	**TRADES**	**MART**	**ATENT** **ℓ_∞**
Benign	**87.30**	84.92	84.17	**85.67**
ℓ_∞ PGD-20	47.04	56.61	**57.39**	**57.23**
(E)	56.80			
ℓ_∞ CW	49.27	**62.67**	54.53	**62.34**
ℓ_∞ DeepFool	-	**58.15**	55.89	**57.21**

The highest two accuracy values in each row are highlighted.

Attacks: For ℓ_2 attacks, we test PGD-40 with 10 random restarts, and CW2 attacks at radius $\epsilon_2 = 2$ for MNIST and PGD-40 and CW2 attacks at $\epsilon_2 = 0.43$ ($\approx \epsilon_\infty = 2/255$) and $\epsilon_2 = 0.5 = 128/255$ for CIFAR10. For ℓ_∞ attacks, we test PGD-20, ℓ_∞CW, DeepFool attacks at radiii $\epsilon_\infty = 0.3$ for MNIST and $\epsilon_\infty = 0.031 = 8/255$ for CIFAR10. We test ATENT at other attack radii in **Supplementary Material**. For implementing the attacks, we use the Foolbox library (Rauber et al., 2017) and the Adversarial Robustness Toolbox (Nicolae et al. (2018)).

Defenses: We compare models trained using: SGD (vanilla), Entropy SGD (Chaudhari et al., 2019), PGD-AT (Madry et al., 2018) with random starts [or PGD-AT(E) with random start, early stopping (Rice et al., 2020)], TRADES (Zhang et al., 2019b), MMA (Ding et al., 2019) and MART (Wang et al., 2019). Wherever available, we use pretrained models to tabulate robust accuracy results for PGD-AT, TRADES, MMA and MART as presented in their published versions. Classifiers giving the best and second best accuracies are highlighted in each category. We note here that a good defense mechanism should give better robust accuracies across different attack strategies. Since no single defense strategy outperforms every other defenses across all attacks, we highlight the *best two* robust accuracies. We find that ATENT obtains either the best or second best robust accuracy against all methods tested. This suggests that ATENT generalizes better than other defense strategies against various attacks.

Smoothing: We also test randomized smoothing (Cohen et al., 2019) in addition to our adversarial training to evaluate certified robust accuracies.

The results were generated using an Intel(R) Xeon(R) W-2195 CPU 2.30 GHz Lambda cluster with 18 cores and a NVIDIA TITAN GPU running PyTorch version 1.4.0.

4.1 MNIST

In **Tables 1** and **2**, we tabulate the robust accuracy for 5-layer convolutional network trained using the various approaches discussed above for both ℓ_2 and ℓ_∞ attacks respectively.

Training setup: Complete details are provided in **Supplementary Material**. Our experiments for ℓ_2 attack are presented in **Table 1**. We perform these experiments on a LeNet5 model imported from the Advertorch toolbox (architecture details are provided in the supplement). For ℓ_2-ATENT we use a batch-size of 50 and SGD with learning rate of $\eta = 0.001$ for updating weights. We set $\gamma = 0.05$ and noise

$\varepsilon \sim 0.001\mathcal{N}(0, \mathbb{I})$. We perform $K = 40$ Langevin epochs and set the Langevin parameter $\alpha = 0.9$, and step $\eta' = 0.25$. For attack, we do a 40-step PGD attack with ℓ_2-ball radius of $\epsilon = 2$. The step size for the PGD attack is 0.25, consistent with the configuration in Ding et al. (2019). We perform early stopping by tracking robust accuracies of validation set and report the best accuracy found.

In **Table 2**, we use a SmallCNN configuration as described in Zhang et al. (2019b) (architecture in supplement). We use a batch-size of 128, SGD optimizer with learning rate of $\eta = 0.01$ for updating weights. We set $\gamma = 3.33$ and noise $\varepsilon \sim 0.001\mathcal{N}(0, \mathbb{I})$. We perform $L = 40$ Langevin epochs and we set the Langevin parameter $\alpha = 0.9$, and step $\eta' = 0.01$, consistent with the configuration in Zhang et al. (2019b). For the PGD attack, we use a 20-step PGD attack with step-size 0.01, for ℓ_∞-ball radius of $\epsilon = 0.3$. We perform an early stopping by tracking robust accuracies on the validation set and report the best accuracy found. Other attack configurations can be found in the supplement.

Our experiments on the Entropy-SGD (row 2 in **Tables 1** and **2**) trained network suggests that networks trained to find flat minima (with respect to weights) are not more robust to adversarial samples as compared to vanilla SGD.

4.2 CIFAR10

Next, we extend our experiments to CIFAR-10 using a WideResNet 34-10 as described in Zhang et al. (2019b); Wang et al. (2019) as well as ResNet-20. For PGD-AT (and PGD-AT (E)), TRADES, and MART, we use the default values stated in their corresponding papers.

Training setup: Complete details in **Supplementary Material**. Robust accuracies of WRN-34-10 classifer trained using state of art defense models are evaluated at the ℓ_∞ attack benchmark requirement of radius $\epsilon = 8/255$, on CIFAR10 dataset and tabulated in **Table 3**. For ℓ_∞-ATENT, we use a batch-size of 128, SGD optimizer for weights, with learning rate $\eta = 0.1$ (decayed to 0.01 at epoch 76), 76 total epochs, weight decay of 5×10^{-4} and momentum 0.9. We set $\gamma = 1/(0.0031)$, $K = 10$ Langevin iterations, $\varepsilon = 0.001\mathcal{N}(0, \mathbb{I})$, at step size $\eta' = 0.007$. We test against 20-step PGD attack, with step size 0.003, as well as ℓ_∞-CW and Deep Fool attacks using FoolBox. ℓ_∞-ATENT is consistently among the top two performers at benchmark configurations.

Importance of early stopping: Because WRN34-10 is highly overparameterized with approximately 48 million trainable parameters, it tends to overfit adversarially-perturbed CIFAR10 examples. The success of TRADES (and also PGD) in Rice et al. (2020) relies on an early stopping condition and corresponding learning rate scheduler. We strategically search different early stopping points and report the best possible robust accuracy from different stopping points.

We test efficiency of our ℓ_2-based defense on both ℓ_2 attacks, as well as compute ℓ_2 certified robustness for the smoothed version of ATENT against smoothed TRADES (Blum et al., 2020) in **Supplementary Table S2** in **Supplementary Material**. We find that our formulation of ℓ_2 ATENT is both robust against ℓ_2 attacks, as well as gives a competitive certificate against adversarial perturbations for ResNet20 on CIFAR10.

In **Supplementary Material** we also demonstrate a fine-tuning approach for ATENT, where we consider a pre-trained WRN34-10 and fine tune it using ATENT, similar to the approach in Jeddi et al. (2020). We find that ATENT can be used to fine tune a naturally pretrained model at lower computational complexity to give competitive robust accuracies while almost retaining the performance on benign data.

4.3 Discussion

We propose a new algorithm for defending neural networks against adversarial attacks. We demonstrate competitive (and often improved) performance of our family of algorithms (ATENT) against the state of the art. We analyze the connections of ATENT with both PGD-adversarial training as well as randomized smoothing. Future work includes extending to larger datasets such as ImageNet, as well as theoretical analysis for algorithm convergence.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding authors.

AUTHOR CONTRIBUTIONS

GJ contributed to probablistic modeling, design of the algorithm and conducted some of the experiments. AJ added to the experimental analysis by designing and performing suitable baseline comparisons. AC performed the initial experimental analysis and researched the code for baseline models. GJ contributed to writing all sections in the first draft of the manuscript. AJ added experimental sections of the manuscript and AC added literature review. SG and CH gave feedback on the problem formulation and suggested papers to be added to literature survey as well as helped design experimental comparisons. All authors contributed to manuscript revision, read, and approved the submitted version.

FUNDING

This work was supported in part by NSF grant awards CCF-2005804 and CCF-1815101, and NSF/USDA-NIFA award 2021-67021-35329.

ACKNOWLEDGMENTS

The authors would like to thank Anna Choromanska for useful initial feedback and Praneeth Narayanamurthy for insightful conceptual discussions.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frai.2021.780843/full#supplementary-material

REFERENCES

Athalye, A., Carlini, N., and Wagner, D. (2018). *Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples*.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., and Laskov, P. (2013). "Evasion Attacks against Machine Learning at Test Time," in Joint European conference on machine learning and knowledge discovery in databases (Springer), 387–402. doi:10.1007/978-3-642-40994-3_25

Biggio, B., and Roli, F. (2018). Wild Patterns: Ten Years after the Rise of Adversarial Machine Learning. *Pattern Recognition* 84, 317–331. doi:10.1016/j.patcog.2018.07.023

Blum, A., Dick, T., Manoj, N., and Zhang, H. (2020). Random Smoothing Might Be Unable to Certify L∞ Robustness for High-Dimensional Images. *J. Machine Learn. Res.* 21, 1–21.

Carlini, N., Mishra, P., Vaidya, T., Zhang, Y., Sherr, M., Shields, C., et al. (2016). "Hidden Voice Commands," in 25th {USENIX} Security Symposium ({USENIX} Security 16, 513–530.

Carlini, N., and Wagner, D. A. (2017b). "Towards Evaluating the Robustness of Neural Networks," in 2017 IEEE Symposium on Security and Privacy (SP). doi:10.1109/sp.2017.49

Carlini, N., and Wagner, D. (2017a). "Towards Evaluating the Robustness of Neural Networks," in 2017 IEEE Symposium on Security and Privacy (IEEE), 39–57. doi:10.1109/sp.2017.49

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., et al. (2019). Entropy-sgd: Biasing Gradient Descent into Wide Valleys. *J. Stat. Mech. Theor. Exp.* 2019, 124018. doi:10.1088/1742-5468/ab39d9

Cohen, J., Rosenfeld, E., and Kolter, Z. (2019). "Certified Adversarial Robustness via Randomized Smoothing," in Proceedings of the 36th International Conference on Machine Learning. Editors K. Chaudhuri and R. Salakhutdinov, and California Long Beach (Long Beach, CA: Proceedings of Machine Learning Research), USA: PMLR, 97, 1310–1320.

Croce, F., and Hein, M. (2020). "Reliable Evaluation of Adversarial Robustness with an Ensemble of Diverse Parameter-free Attacks," in International conference on machine learning (PMLR), 2206–2216.

Dathathri, S., Zheng, S., Gao, S., and Murray, R. (2017). "Measuring the Robustness of Neural Networks via Minimal Adversarial Examples," in Deep Learning: Bridging Theory and Practice, NIPS 2017 workshop, Long Beach, CA 35 (NeurIPS-W).

Ding, G., Sharma, Y., Lui, K., and Huang, R. (2019). "Mma Training: Direct Input Space Margin Maximization through Adversarial Training," in International Conference on Learning Representations.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. (2017). "Sharp Minima Can Generalize for Deep Nets," in nternational Conference on Machine Learning, 1019–1028.*I*.

Fan, Y., Wu, B., Li, T., Zhang, Y., Li, M., Li, Z., et al. (2020). "Sparse Adversarial Attack via Perturbation Factorization," in Computer Vision–ECCV 2020: 16th European ConferenceProceedings, Part, Glasgow, UK, August 23–28, 2020 (Springer), 35–50. doi:10.1007/978-3-030-58542-6_3XXII 16

Goodfellow, I. J. (2018). *Defense against the Dark Arts: An Overview of Adversarial Example Security Research and Future Research Directions*. *arxiv preprint* abs/1806.04169.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). *Explaining and Harnessing Adversarial Examples*. arXiv preprint arXiv:1412.6572.

Goodfellow, I., Shlens, J., and Szegedy, C. (2015). "Explaining and Harnessing Adversarial Examples," in ICLR.

Gowal, S., Dvijotham, K., Stanforth, R., Bunel, R., Qin, C., Uesato, J., et al. (2019). "Scalable Verified Training for Provably Robust Image Classification," in 2019 IEEE/CVF International Conference on Computer Vision (ICCV) (IEEE), 4841–4850. doi:10.1109/iccv.2019.00494

Hochreiter, S., and Schmidhuber, J. (1997). Flat Minima. *Neural Comput.* 9, 1–42. doi:10.1162/neco.1997.9.1.1

Jeddi, A., Shafiee, M., and Wong, A. (2020). *A Simple fine-tuning Is All You Need: Towards Robust Deep Learning via Adversarial fine-tuning*. arXiv preprint arXiv:2012.13628.

Jiang, Z., Chen, T., Chen, T., and Wang, Z. (2020). "Robust Pre-training by Adversarial Contrastive Learning," in *NeurIPS*.

Joshi, A., Jagatap, G., and Hegde, C. (2021). *Adversarial Token Attacks on Vision Transformers*. arXiv preprint arXiv:2110.04337.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. (2016). *On Large-Batch Training for Deep Learning: Generalization gap and Sharp Minima*. arXiv preprint arXiv:1609.04836.

Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). "Imagenet Classification with Deep Convolutional Neural Networks," in Advances in Neural Information Processing Systems, 1097–1105.

Kurakin, A., Goodfellow, I., and Bengio, S. (2016). *Adversarial Examples in the Physical World*. arXiv preprint arXiv:1607.02533.

Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S. (2019). "Certified Robustness to Adversarial Examples with Differential Privacy," in 2019 IEEE Symposium on Security and Privacy (SP) (IEEE), 656–672. doi:10.1109/sp.2019.00044

Li, B., Chen, C., Wang, W., and Carin, L. (2018a). *Certified Adversarial Robustness with Additive Gaussian Noise*. arXiv preprint arXiv:1809.03113.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. (2018b). "Visualizing the Loss Landscape of Neural Nets," in Advances in Neural Information Processing Systems, 6389–6399.

Li, Y., Min, M. R., Lee, T., Yu, W., Kruus, E., Wang, W., et al. (2021). "Towards Robustness of Deep Neural Networks via Regularization," in Proceedings of the IEEE/CVF International Conference on Computer Vision(ICCV), 7496–7505.

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

Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., and Frossard, P. (2017). "Universal Adversarial Perturbations," in IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI (CVPR), 1765–1773.

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. (2016). "Deepfool: A Simple and Accurate Method to Fool Deep Neural Networks," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (Las Vegas, NV: CVPR). doi:10.1109/cvpr.2016.282

Nicolae, M.-I., Sinn, M., Tran, M. N., Buesser, B., Rawat, A., Wistuba, M., et al. (2018). *Adversarial Robustness Toolbox v1.2.0*. arXiv preprint arXiv:1807.01069.

Papernot, N., McDaniel, P., and Goodfellow, I. (2016). *Transferability in Machine Learning: From Phenomena to Black-Box Attacks Using Adversarial Samples*. arXiv preprint arXiv:1605.07277.

Paul, S., and Chen, P.-Y. (2021). *Vision Transformers Are Robust Learners*. arXiv preprint arXiv:2105.07581.

Rauber, J., Brendel, W., and Bethge, M. (2017). *Foolbox: A python Toolbox to Benchmark the Robustness of Machine Learning Models*. arXiv preprint arXiv:1707.04131.

Ren, K., Zheng, T., Qin, Z., and Liu, X. (2020). Adversarial Attacks and Defenses in Deep Learning. *Engineering* 6 (3), 346–360. doi:10.1016/j.eng.2019.12.012

Rice, L., Wong, E., and Kolter, Z. (2020). "Overfitting in Adversarially Robust Deep Learning," in International Conference on Machine Learning (PMLR), 8093–8104.

Rony, J., Granger, E., Pedersoli, M., and Ben Ayed, I. (2021). "Augmented Lagrangian Adversarial Attacks," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 7738–7747.

Rota, P., Sangineto, E., Conotter, V., and Pramerdorfer, C. (2016). "Bad Teacher or Unruly Student: Can Deep Learning Say Something in Image Forensics Analysis," in 2016 23rd International Conference on Pattern Recognition (ICPR) (IEEE), 2503–2508.

Salman, H., Li, J., Razenshteyn, I., Zhang, P., Zhang, H., Bubeck, S., et al. (2019a). "Provably Robust Deep Learning via Adversarially Trained Smoothed Classifiers," in Advances in Neural Information Processing Systems, 11289–11300.

Salman, H., Li, J., Razenshteyn, I., Zhang, P., Zhang, H., Bubeck, S., et al. (2019b). Provably Robust Deep Learning via Adversarially Trained Smoothed Classifiers. *Adv. Neural Inf. Process. Syst.* 32, 11292–11303.

Shafahi, A., Najibi, M., Ghiasi, M. A., Xu, Z., Dickerson, J., Studer, C., et al. (2019). "Adversarial Training for Free," in Advances in Neural Information Processing Systems, 3353–3364.

Shao, R., Shi, Z., Yi, J., Chen, P.-Y., and Hsieh, C.-J. (2021). *On the Adversarial Robustness of Visual Transformers*. arXiv preprint arXiv:2103.15670.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., et al. (2013). *Intriguing Properties of Neural Networks*. arXiv preprint arXiv: 1312.6199.

Tang, J., Du, X., He, X., Yuan, F., Tian, Q., and Chua, T. (2019). “Adversarial Training towards Robust Multimedia Recommender System,” in *IEEE Transactions on Knowledge and Data Engineering*.

Tjeng, V., Xiao, K., and Tedrake, R. (2018). “Evaluating Robustness of Neural Networks with Mixed Integer Programming,” in International Conference on Learning Representations.

Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. (2017). *Ensemble Adversarial Training: Attacks and Defenses*. arXiv preprint arXiv:1705.07204.

Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., and Gu, Q. (2019). “Improving Adversarial Robustness Requires Revisiting Misclassified Examples,” in International Conference on Learning Representations.

Welling, M., and Teh, Y. (2011). “Bayesian Learning via Stochastic Gradient Langevin Dynamics,” in Proceedings of the 28th International Conference on Machine Learning (Bellevue, Washington D.C: ICML), 681–688.

Wong, E., and Kolter, Z. (2018). “Provable Defenses against Adversarial Examples via the Convex Outer Adversarial Polytope,” in International Conference on Machine Learning (Stockholm, Sweden: PMLR), 5286–5295.

Wong, E., Rice, L., and Kolter, J. Z. (2020). *Fast Is Better than Free: Revisiting Adversarial Training*.

Wong, E., Schmidt, F., Metzen, J., and Kolter, J. (2018). “Scaling Provable Adversarial Defenses,” in *NeurIPS*.

Xu, P., Chen, J., Zou, D., and Gu, Q. (2017). *Global Convergence of Langevin Dynamics Based Algorithms for Nonconvex Optimization*. arXiv preprint arXiv: 1707.06618.

Zhang, H., Chen, H., Xiao, C., Gowal, S., Stanforth, R., Li, B., et al. (2019a). “Towards Stable and Efficient Training of Verifiably Robust Neural Networks,” in International Conference on Learning Representations.

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. (2019b). “Theoretically Principled Trade-Off between Robustness and Accuracy,” in International Conference on Machine Learning, 7472–7482.

Zhang, W., Sheng, Q., Alhazmi, A., and Li, C. (2020). “Adversarial Attacks on Deep-Learning Models in Natural Language Processing: A Survey,” in *ACM Transactions on Intelligent Systems and Technology* (New York, NY: Association for Computing Machinery), 11, 1–41. doi:10.1145/3374217

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher’s Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.