Adversarial Purification with Score-based Generative Models

Jongmin Yoon [1] **Sung Ju Hwang** [1 2] **Juho Lee** [1 2]

Abstract

While adversarial training is considered as a standard defense method against adversarial attacks for image classifiers, adversarial purification, which purifies attacked images into clean images with a standalone purification model, has shown promises as an alternative defense method. Recently, an Energy-Based Model (EBM) trained with Markov-Chain Monte-Carlo (MCMC) has been highlighted as a purification model, where an attacked image is purified by running a long Markov-chain using the gradients of the EBM. Yet, the practicality of the adversarial purification using an EBM remains questionable because the number of MCMC steps required for such purification is too large. In this paper, we propose a novel adversarial purification method based on an EBM trained with Denoising Score-Matching (DSM). We show that an EBM trained with DSM can quickly purify attacked images within a few steps. We further introduce a simple yet effective randomized purification scheme that injects random noises into images before purification. This process screens the adversarial perturbations imposed on images by the random noises and brings the images to the regime where the EBM can denoise well. We show that our purification method is robust against various attacks and demonstrate its state-of-the-art performances.

1. Introduction

Image classifiers built with deep neural networks are known to be vulnerable to adversarial attacks, where an image containing a small perturbation imperceptible to human completely changes the prediction results (Goodfellow et al., 2015; Kurakin et al., 2017). There are various methods that aim to make classifiers robust to such adversarial attacks, and *adversarial training* (Madry et al., 2018; Zhang et al.,

2019), in which a classifier is trained with adversarial examples, is considered as a standard defense method due to its effectiveness.

Another approach for the adversarial defense is to *purify* attacked images before feeding them to classifiers. This strategy, referred to as *adversarial purification* (Srinivasan et al., 2021; Yang et al., 2019; Shi et al., 2021), learns a *purification model* whose goal is to remove any existing adversarial noise from potentially attacked images into clean images so that they could be correctly classified when fed to the classifier. The purification model is usually trained independently of the classifier and does not necessarily require the class labels for training; thus, it is less likely to affect the clean image classification accuracy compared to the adversarial training methods. The most common way is for adversarial purification to learn a *generative model* over the images (Samangouei et al., 2018; Song et al., 2018; Schott et al., 2019; Ghosh et al., 2019) such that one can restore clean images from attacked images.

Recently, along with the advances in learning Energy-Based Models (EBMs) with deep neural networks, adversarial purification methods using an EBM trained with Markov-Chain Monte-Carlo (MCMC) as a purification model have been highlighted (Du & Mordatch, 2019; Grathwohl et al., 2020; Hill et al., 2021). Hinging on the memoryless behavior of MCMC, these methods purify attacked images by running a large number of sampling steps defined by Langevin dynamics. When started from an attacked image, in the long run, Langevin sampling will eventually bring the attacked image to a clean image that is likely to be generated from the data distribution. However, the success of the purification heavily depends on the number of sampling steps, and unfortunately, the number of steps required for stably purify attacked images is too large to be practical.

In this paper, we propose a novel adversarial purification method using an EBM trained with Denoising Score-Matching (DSM) (Hyvärinen, 2005; Song & Ermon, 2019) as a purification model. Unlike an EBM trained with MCMC estimating the energy function, DSM (Vincent et al., 2010) learns the *score function* that can *denoise* the perturbed samples, which is more closely related to the adversarial purification because the purification can be thought as denoising of the adversarial attacks. We show that an EBM trained with DSM, using the *deterministic update scheme*

---

[1]Korea Advanced Institute of Science and Technology, Daejeon, Korea [2]AITRICS, Seoul, Korea. Correspondence to: Jongmin Yoon <jm.yoon@kaist.ac.kr>, Juho Lee <juholee@kaist.ac.kr>.

that we propose, can quickly purify the attacked images within several orders of magnitude fewer steps than the previous methods. We further propose a simple technique to enhance the robustness of our purification method by injecting noises to images before the purification. The intuition behind this is, by injecting noises relatively larger than adversarial perturbations, we can make the adversarial perturbations negligible and also at the same time convert the images to the familiar noisy images that are similar to the ones seen during the training with DSM. As our model is facilitated by random noises around the images, the classifier with randomly purified images can be interpreted as a *randomized smoothing classifier* (Cohen et al., 2019; Pinot et al., 2020). Since the noise distribution over clean images and attacked images overlap, we validate that our denoising framework delivers *certified robustness* over any threat models, implying that the attacked image is guaranteed to be provably robust regardless of the attacks driven.

We verify our method on various adversarial attack benchmarks and demonstrate its state-of-the-art performance. Unlike the previous purification methods, we validate our defense method using a strong attack that approximates the gradients of the entire purification procedure. We show the state-of-the-art performance of our methods on several datasets using various attacks.

2. Backgrounds

2.1. Adversarial training, preprocessing, and purification

Consider a classifier neural network $g_\phi : \mathbb{R}^D \to \mathbb{R}^K$. A neural network robust to adversarial attacks can be trained with the *worst-case risk minimization*,

$$\min_\phi \mathbb{E}_{p_{\text{data}}(x,y)} \left[\max_{x' \in \mathcal{B}(x)} \mathcal{L}\left(g_\phi\left(x'\right), y\right) \right], \quad (1)$$

where $\mathcal{L}$ is a loss function and $\mathcal{B}(x)$ is the threat model at x (i.e., the domain of all attacked images from x). The exact evaluation and optimization of this objective are infeasible, so one must resort to an approximation. A predominant approach is *adversarial training* (Madry et al., 2018; Zhang et al., 2019; Carmon et al., 2019), which optimizes a surrogate loss computed from restricted classes of adversarial examples during training. For instance, Madry et al. (2018) approximates the inner maximization loop in (1) with the average loss computed for the chosen attacked images.

$$\min_\phi \mathbb{E}_{p_{\text{data}}(x,y)} \left[\mathbb{E}_{x' \in \mathcal{A}(x)} \mathcal{L}\left(g_\phi\left(x'\right), y\right) \right], \quad (2)$$

where $\mathcal{A}(x_i)$ is the set of images obtained by attacking x with chosen classes of attacks. Due to the relaxation of the full threat model, the resulting network may still be vulnerable to attacks not included in $\mathcal{A}(x)$.

Another approach for adversarial defense is *preprocessing*, where input images are preprocessed with auxiliary transformations before classification. Let f_θ be a preprocessor transforming images. The training objective is then

$$\min_{\phi, \theta} \mathbb{E}_{p_{\text{data}}(x,y)} \left[\max_{x' \in \mathcal{B}(x)} \mathcal{L}(g_\phi(f_\theta(x')), y) \right]. \quad (3)$$

The maximum over the threat model $\mathcal{B}(x)$ may be approximated with an average over known classes of adversarial attacks or average over stochastically transformed inputs. Such transformations include adding stochasticity into the input images or adding discrete or non-differentiable transforms into the input images (Guo et al., 2018; Dhillon et al., 2018; Buckman et al., 2018; Xiao et al., 2020), making the gradient estimation with respect to the loss function $\nabla_x \mathcal{L}(g_\phi(f_\theta(x)), y)$ harder for an attacker.

Finally, in *adversarial purification*, a *generative model* that can restore clean images from attacked images is additionally trained and used as a preprocessor (Samangouei et al., 2018; Song et al., 2018; Srinivasan et al., 2021; Ghosh et al., 2019; Hill et al., 2021; Shi et al., 2021), where the preprocessor f_θ corresponds to the purification process defined with the generative model.

2.2. Energy-based models and adversarial purification

An EBM (Lecun et al., 2006) defined on $\mathbb{R}^D$ is a probabilistic model whose density function is written as

$$p_\theta(x) = \frac{\exp(-E_\theta(x))}{Z_\theta}, \quad (4)$$

where $E_\theta(x) : \mathbb{R}^D \to \mathbb{R}$ is the *energy function* and $Z_\theta = \int_x \exp(-E_\theta(x))\mathrm{d}x$ is the normalization constant. Since $E_\theta(x)$ is not subject to any constraint (e.g., integrate to one), an EBM provides great flexibility in choosing the form of the model. However, due to the intractable normalization constant, computing the density and learning the parameter θ requires approximation. Roughly speaking, there are three methods to learn an EBM; maximum-likelihood with MCMC, score matching (Hyvärinen, 2005), and noise-contrastive estimation (Gutmann & Hyvärinen, 2010).

Training an EBM with maximum likelihood involves the computation of the gradient

$$\nabla_\theta \mathbb{E}_{p_{\text{data}}(x)}[\log p_\theta(x)]$$
$$= \mathbb{E}_{p_\theta(x)}[\nabla_\theta E_\theta(x)] - \mathbb{E}_{p_{\text{data}}(x)}[\nabla_\theta E_\theta(x)]. \quad (5)$$

Here, the first term evaluates the intractable expectation over the model distribution $p_\theta(x)$ and is usually approximated with Monte-Carlo approximation. Recently, drawing samples from $p_\theta(x)$ from a Markov-chain defined with Langevin dynamics has been demonstrated to work well, even for high-dimensional energy functions constructed with deep

neural networks (Nijikamp et al., 2019; Du & Mordatch, 2019). Using Langevin sampling, one can simulate $p_\theta(x)$ by running a Markov chain,

$$x_t = x_{t-1} - \frac{\alpha}{2}\nabla_x E_\theta(x_{t-1}) + \sqrt{\alpha}\varepsilon, \qquad (6)$$

where $\alpha > 0$ is a step size, x_0 is a randomly initialized starting point, and $\varepsilon \sim \mathcal{N}(0, I)$. This can be used not only for the training but also for generating novel samples from trained EBMs. Based on this, Du & Mordatch (2019); Grathwohl et al. (2020) showed that an EBM trained with MCMC can purify adversarially attacked images by running Langevin sampling starting from the attacked images. Hill et al. (2021) further developed this by adjusting the optimization process of an EBM to make it better convergent for long MCMC chains, and demonstrated that the adversarial purification with long-run Langevin sampling can successfully purify attacked images into clean images. The key for the successful purification is the "long-run" MCMC sampling. The number of MCMC steps required for purification is typically more than 1,000, which is costly even with modern GPUs.

2.3. Denoising score matching

Score-Matching (SM) (Hyvärinen, 2005) is a density estimation technique that learns the *score function* of the target density instead of directly learning the density itself. Let $p_{\text{data}}(x)$ be a target density defined on $\mathbb{R}^D$, and $\log p_\theta(x)$ be a model. The score model $\nabla_x \log p_\theta(x) := s_\theta(x) : \mathbb{R}^D \to \mathbb{R}^D$ is then trained to approximate the true score function $\nabla_x \log p_{\text{data}}(x)$ by minimizing the objective $\mathbb{E}_{p_{\text{data}}(x)}[\frac{1}{2}\|s_\theta(x) - \nabla_x \log p_{\text{data}}(x)\|]$, and under mild conditions, this can be shown to be equivalent to minimizing

$$\mathbb{E}_{p_{\text{data}}(x)}\left[\frac{1}{2}\|s_\theta(x)\|^2 + \text{tr}(\nabla_x s_\theta(x))\right]. \qquad (7)$$

Hyvärinen (2005) showed that this objective gives a consistent estimator of the true parameter θ. SM is useful for training an EBM because computing the score functions does not require computing the intractable normalizing constants. However, still, the basic version of score matching with objective (7) does not scale to high-dimensional data due to the term $\text{tr}(\nabla_x s_\theta(x))$.

To avoid computing the term $\text{tr}(\nabla_x s_\theta(x))$, DSM (Vincent, 2011) slightly tweaks the objective (7). The basic idea of DSM is to learn the score function of the *perturbed* data. Given a pre-specified noise distribution $q(\tilde{x}|x)$, DSM minimizes the following objective,

$$\mathbb{E}_{q(\tilde{x}|x)p_{\text{data}}(x)}\left[\frac{1}{2}\|s_\theta(\tilde{x}) - \nabla_{\tilde{x}} \log q(\tilde{x}|x)\|^2\right]. \qquad (8)$$

This modified objective is well-defined, provided that the noise distribution is smooth, and if the noise is small so
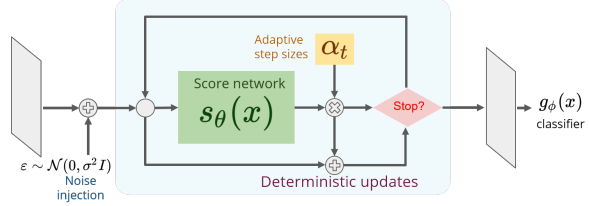

Figure 1. A conceptual diagram describing our approach.

that $q(x) := \int q(x|x')p_{\text{data}}(x')\mathrm{d}x' \approx p_{\text{data}}(x)$, DSM finds the same solution as the original SM. A common choice for $q(\tilde{x}|x)$ is the Gaussian distribution centered at x, $q(\tilde{x}|x) = \mathcal{N}(\tilde{x}; x, \sigma^2 I)$. In such case, (8) reduces to

$$\ell(\theta, \sigma) = \mathbb{E}_{q(\tilde{x}|x)p_{\text{data}}(x)}\left[\frac{1}{2\sigma^4}\|\tilde{x} + \sigma^2 s_\theta(\tilde{x}) - x\|^2\right]. \qquad (9)$$

That is, $s_\theta(\tilde{x})$ is trained to recover the original data x from the perturbed data $\tilde{x}$, as in denoising autoencoders (Vincent et al., 2008; 2010).

3. Adversarial purification with score-based generative models

3.1. Denoising score matching for adversarial purification

While previous adversarial purification methods using EBMs employ maximum likelihood training with MCMC, we propose to use DSM to train an EBM to be used for purification. As we discussed above, DSM aims to learn a score network $s_\theta(x)$ pointing the direction of restoring original samples from perturbed samples. Considering that the adversarial purification can be understood as a denoising procedure, we conjecture that DSM is a better option to train an EBM as it is learned with the objective more directly related to the adversarial purification. The maximum likelihood training aims to learn the energy function $E_\theta(x)$, and the gradients $\nabla_x E_\theta(x)$ is obtained as a byproduct. Hence, unless trained perfectly, accurately predicting the energy function does not necessarily mean accurately predicting the gradients $\nabla_x E_\theta(x)$ that are actually used for purification.

In this work, we adopt the recently proposed Noise Conditional Score Network (NCSN) (Song & Ermon, 2019; 2020) for our model. An NCSN is trained with a modified DSM incorporating multiple noise levels, where inputs are perturbed with multiple noise levels $\{\sigma_j\}_{j=1}^L$ instead of a single noise σ. The training objective based on this multi-scale noise is

$$\mathcal{L}(\theta, \{\sigma_j\}_{j=1}^L) = \sum_{j=1}^L \sigma_j^2 \ell(\theta, \sigma_j), \qquad (10)$$

where $\ell(\theta, \sigma_j)$ is the DSM objective (9) with a specific noise
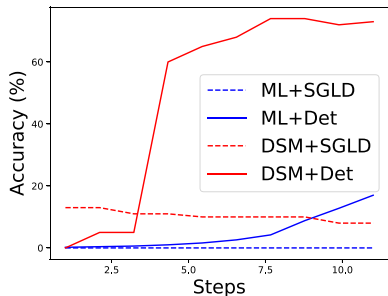
Figure 2. The accuracy against the BPDA attack on CIFAR10. ML denotes the maximum likelihood training with MCMC, and Det denotes deterministic updates.

level. We choose the noise levels $\{\sigma_j\}_{j=1}^L$ following the guidance in Song & Ermon (2020). Training with multiple noises exposes the score network $s_\theta(x)$ to the various perturbed samples, and this may be advantageous for the adversarial purification of various attacked images.

Song & Ermon (2020) observed that the norm of the noise-conditioned score function trained with Eq. (10) is approximately reciprocal to the noise level, i.e., $\|s_\theta(x,\sigma)\|_2 \propto 1/\sigma$, and proposed to use the conditional score function as $s_\theta(x,\sigma) = s_\theta(x)/\sigma$. Throughout the paper, we follow this parameterization for our purification model.

3.2. Purification by deterministic updates

Let $s_\theta(x)$ be a trained score network and g_ϕ be a classifier network with softmax output. Given an attacked image x', previous adversarial purification methods would run a stochastic Markov chain driven by Langevin dynamics, with $s_\theta(x)$ in place of $-\nabla_x E_\theta(x)$. Instead, we propose to *deterministically update* the samples with the learned scores. That is, starting from $x_0 = x'$, execute the following updates for $t \geq 1$,

$$x_t = x_{t-1} + \alpha_{t-1} s_\theta(x_{t-1}), \qquad (11)$$

where $\{\alpha_t\}_{t \geq 0}$ are step-sizes that may be tuned with a validation set or chosen adaptively using the algorithm that we will describe in short.

While this deterministic update does not guarantee convergence, we empirically found that it improves classification accuracy much faster than stochastic updates. We also found that the deterministic update also slightly improves the speed of purification for EBM trained with MCMC in the short run, but DSM works much better with deterministic updates (Fig. 2).

After running T steps of deterministic updates, we can pass x_T through the classifier to get a prediction. We may stop the iteration before T steps when the adaptive step size α_t

becomes less then the pre-specified threshold τ.

3.3. Random noise injection before purification

As we will show in Section 5, the defense method based on the deterministic purification described in Section 3.2 can successfully defend most of the adversarial attacks, but it is vulnerable to the strong attack based on the gradient estimation of the full purification process. We propose a simple enhancement to our algorithm that can even defend this strong attack and improve classification accuracy. The idea is simple; We add a random Gaussian noises to images before purification. Given a potentially attacked image x', the purification proceeds as

$$x_0 = x' + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2 I)$$
$$x_t = x_{t-1} + \alpha_{t-1} s_\theta(x_{t-1}). \qquad (12)$$

The intuition behind this is as follows. Assume that an attacked image x' contains an adversarial perturbation as $x' = x + \nu$, and we add a Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$ to form $x_0 = x + \nu + \varepsilon$. Since the norm of ν is bounded due to the perceptual indistinguishability constraint, the added noise ε can "screen out" the relatively small perturbation ν. Also, recall that the score network $s_\theta(x)$ is trained to denoise images perturbed by Gaussian noises. Adding Gaussian noises makes x_0 more similar to the data used to train the score network.

The initial noise level σ is an hyperparameter to be specified. Following the popular heuristic used for kernel methods (Garreau et al., 2018), we choose it as the median of the pairwise Euclidean distances divided by the square root of the input dimension $\sqrt{D}$. The initial noise level selected with this heuristic is $\sigma = 0.25$ for both CIFAR-10 and CIFAR-100.

Due to the randomness of the purification induced by noise injection, we execute the multiple purification runs and take the ensemble as

$$\hat{y} = \underset{k \in [K]}{\operatorname{argmax}} \frac{1}{S} \sum_{s=1}^{S} [g_\phi(x_T^{(s)})]_k, \qquad (13)$$

where $\{x_t^{(s)}\}_{t=1}^T$ is an instance of a purification using (12). Note that the computation for multiple runs of purification (along with s indices) can be parallelized.

Injecting noises before purification now turns our deterministic update updates into a random purification method, and thus the classifier taking randomly purified image can be interpreted as a randomized smoothing classifier (Cohen et al., 2019; Pinot et al., 2020). In Section 5, we actually show that the randomized smoothing classifier derived from our method has certified robustness over any norm-bounded threat models.

3.4. Adaptive step sizes

The step-sizes $\{\alpha_t\}_{t \geq 0}$ are important hyperparameters that can affect the purification performance. While these can be tuned with additional vaildation set [1], we propose a simple yet effective adaptation scheme that can choose proper step-sizes during the purification.

Let x_t be an intermediate point during purification. If trained properly, in a small local neighbor of x_t, there exists σ_t such that $s_\theta(x, \sigma_t)$ is close to the score function of a Gaussian distribution $\mathcal{N}(\mu, \sigma_t^2 I)$ for some μ and σ_t. That is,

$$s_\theta(x_t) \approx -\frac{x_t - \mu}{\sigma_t}. \qquad (14)$$

Let α_t be a step size for x_t. We want the score of the updated point $x_{t+1} = x_t + \alpha_t s_\theta(x_t)$ to have a decreased scale with ratio $(1 - \lambda)$ for some $\lambda \in (0, 1)$, since the score decreases to zero as we move x_t closer to the local optimum (μ).

$$s_\theta(x_{t+1}) = (1 - \lambda)s_\theta(x_t), \qquad (15)$$

and this leads to

$$\alpha_t s_\theta(x_t) \approx \lambda \sigma_t s_\theta(x_t) \Rightarrow \alpha_t = \lambda \sigma_t. \qquad (16)$$

Now, to estimate σ_t, we move x_t along the direction of $s_\theta(x)$ by a small step-size δ to compute $x' = x_t + \delta s_\theta(x_t)$. Then σ_t can be approximated as

$$s_\theta(x') - s_\theta(x_t) \approx \frac{-\delta s_\theta(x_t)}{\sigma_t}$$

$$\Rightarrow \sigma_t \approx \frac{\delta \|s_\theta(x_t)\|^2}{\|s_\theta(x_t)\|^2 - s_\theta(x')^\top s_\theta(x_t)}. \qquad (17)$$

Hence, we get

$$\alpha_t = \lambda \delta \left(1 - \frac{s_\theta(x_t)^\top s_\theta(x')}{\|s_\theta(x_t)\|^2} \right)^{-1}, \qquad (18)$$

as our step size. We find this adaptive learning rates works well without much tuning of the parameters λ. For all experiments, we fixed $\lambda = 0.05$ and $\delta = 10^{-5}$.

We call our adversarial purification method combining all the ingredients (DSM + deterministic updates + noise injection + adaptive step sizes) Adaptive Denoising Purification (ADP). The purification procedure with ADP is summarized in Algorithm 1.

4. Related Works

Adversarial training Szegedy et al. (2014); Goodfellow et al. (2015) discovered that visually imperceptible signals

---

[1]When tuning these values on validation set, we choose a single learning rate $\alpha_t = \alpha$ for all $t \geq 0$ for feasibility of search.

Algorithm 1 Adversarial purification with ADP

Input: an input x, the score network s_θ, the classifier g_ϕ, noise scale σ, number of purification runs S, number of steps per each purification run T, adaptive learning rate parameters λ and δ, purification stopping threshold τ.
for $s = 1$ **to** S **do**
  $x_0^{(s)} \leftarrow x + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2 I)$.
  for $t = 1$ **to** T **do**
    $x' \leftarrow x_{t-1}^{(s)} + \delta s_\theta(x_{t-1}^{(s)})$.
    $\alpha_{t-1} \leftarrow \lambda \delta \left(1 - \frac{s_\theta(x_t^{(s)})^\top s_\theta(x')}{\|s_\theta(x_t^{(s)})\|^2} \right)^{-1}$
    $x_t^{(s)} \leftarrow x_{t-1}^{(s)} + \alpha_{t-1} s_\theta(x_{t-1}^{(s)})$.
    if $\left\| s_\theta(x_t^{(s)}) \right\| < \tau$ **then**
      $x_T^{(s)} \leftarrow x_t^{(S)}$.
      break
    end if
  end for
end for
$\hat{y} = \operatorname{argmax}_k \frac{1}{S} \sum_{s=1}^S [g_\phi(x_T^{(s)})]_k$.
Return $\hat{y}$.

---

for humans could effectively disturb the neural network image classifiers. Adversarial training (Kurakin et al., 2017; Madry et al., 2018) learns a robust classifier by augmenting those adversarial examples at the training phase, and has been shown to be the most reliable defense method. Some techniques such as regularization (Zhang et al., 2019; Wang et al., 2020) or self-supervised learning (Carmon et al., 2019) further improves robustness performance.

Preprocessing for adversarial defense Many existing works propose adversarial defense by preprocessing attacked images via auxiliary transformations or stochasticity before classification. They include thermometer encoding (Buckman et al., 2018), stochastic activation pruning (Dhillon et al., 2018), image quilting (Guo et al., 2018), matrix estimation (Yang et al., 2019) and discontinuous activation (Xiao et al., 2020). Due to the transformations, those methods cause the phenomenon so as called *obfuscated gradients* that makes estimating gradients for gradient-based attack difficult, such as shattered gradients or vanishing/exploding gradients. However, Athalye et al. (2018); Tramèr et al. (2020) designed strong attacks that can bring down the robust accuracy of those defense methods to almost zero.

Adversarial purification methods There also have been various adversarial purification methods. Samangouei et al. (2018) proposed defense-GAN which trains a generator restoring clean images from attacked images. Song et al. (2018) showed that an autoregressive generative model can detect and purify adversarial examples. Srinivasan et al. (2021) proposed a purification method with Metropolis-Adjusted Langevin algorithm (MALA) applied to denoising

autoencoders (Vincent et al., 2008). Grathwohl et al. (2020); Du & Mordatch (2019) showed the promise of EBMs trained with MCMC can purify adversarial examples, and Hill et al. (2021) demonstrated that long-run MCMC with EBMs can robustly purify adversarial examples.

5. Experiments

In this section, we validate our defense method ADP from various perspectives. First, we evaluate ADP under strongest existing attacks on ℓ_∞-bounded threat models and compare it to other state-of-the-art defense methods including adversarial training and adversarial purification. Then we show the certified robustness of our model on ℓ_2-bounded threat models and compare it to other existing randomized classifiers. We further verify the perceptual robustness of our method with common corruptions (Hendrycks & Dietterich, 2019) on CIFAR-10. We further validate ours on a variety of datasets including MNIST, FashionMNIST, and CIFAR-100.

For all experiments, we use WideResNet (Zagoruyko & Komodakis, 2016) with depth 28 and width factor 10, having 36.5M parameters. For the score model, we use NCSN having 29.7M parameters. For the purification methods including ours, we use naturally-trained classifier, i.e., we do not use adversarial training or other augmentations. Unless otherwise stated, for ADP, we fixed the adaptive step size parameters $(\lambda, \delta) = (0.05, 10^{-5})$, and computed ensembles over $S = 10$ purification runs, i.e., we take 10 random noise injection over Gaussian distribution $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$, followed by clipping to $[0, 1]$. We fixed the purification stopping threshold τ is given by 0.001. As aforementioned, the noise standard deviation σ was fixed to 0.25 for all experiments, and please refer to the supplementary material for the results with different values of σ. As an ablation, we also tested ADP without noise injection ($\sigma = 0.0$). We found that purification with adaptive step sizes does not work well without the noise injection, so used manually tuned step-size schedule using validation sets. Please refer to the supplementary material for detailed model description and settings. For all of the attacks described later, we fix the threat model to an ℓ_2 ε-ball with $\varepsilon = 8/255$.

5.1. List of adversarial attacks

The full list of attacks we considered is shown in Table 1 with types and updating rules. We briefly describe the attacks in detail.

Preprocessor-blind attack This is the weakest adversarial attack on the list, where an attacker has full access to the classifier but has no access to the purification model. Attacks under this scenario are sometimes considered as *gray-box* attacks in literature, but we consider this as a

transfer-based black-box attack (Uesato et al., 2018) with source model g_ϕ and target model $g_\phi \circ f_\theta$. We test with the Projective Gradient Descent (PGD) attack on the classifier g_ϕ.

Strong adaptive attack Our purification algorithm consists of multiple iterations through neural networks, so might cause obfuscated gradient problems. Hence, we also validate our defense method with strong adaptive attacks, including BPDA (Athalye et al., 2018) and its variants. We test the basic version of BPDA where the purification process $f_\theta(x)$ is approximated with identity function. We also test the following modifications of the BPDA customized to the adversarial purification methods.

- Joint attack (score): updates the input by weighted sum of the classifier gradient and score network output. If an attacked image has low score norm, then it will not be purified by our algorithm.

- Joint attack (full): updates the input by weighted sum of the classifier gradient and difference between an original input and purified input.

Since our defense method contains random noise injection, we validate our defense method with Expectation Over Time (EOT) (Athalye et al., 2018) attacks together with strong adaptive attacks.

Score-based black-box attack Even when an attacker does not have access to a model and its gradient with respect to a loss function, the gradient can still be estimated with large number of samples. One of such approaches is SPSA (Spall, 1987; Uesato et al., 2018), where the random samples near an input are drawn and the approximate gradient is obtained by expected value of gradients approximated with the finite-difference method. One caveat of these attacks is that the number of samples required to estimate gradients can be large. In our setting, we set the number of queries to 1,280 to make the attack strong enough.

Common corruption Hendrycks & Dietterich (2019) proposed *common corruptions*, a class of 75 frequently occuring visual perturbations and suggested to use them for testing robustness of classifiers. We test our defense method on CIFAR-10-C where those common corruptions are applied to CIFAR-10 dataset. While ours and other adversarial defense methods are not designed to defend these perturbations, they can still be a good way to test robustness of defense methods.

5.1.1. EVALUATION: PREPROCESSOR-BLIND ATTACKS

The evaluation of defense methods on CIFAR-10 with preprocessor-blind attacks are shown in Table 2. We present

Table 1. List of attacks considered. After each update, the output is projected with $x_{i+1} = \prod_{\mathcal{B}_\infty(x_0, b)} x'_{i+1}$. Here $f_\theta : \mathbb{R}^D \to \mathbb{R}^D$ is the full purification model, $s_\theta : \mathbb{R}^D \to \mathbb{R}^D$ is the score network that consists the purification and $g_\phi : \mathbb{R}^D \to \mathbb{R}^K$ is the classifier, where D is the dimension of data and K is the number of classes. For SPSA attack, v_i is uniformly sampled from $\{-1, 1\}^D$. For all of our experiments, we fix $\alpha_i = 2/255$ and $\varepsilon = 0.5$.

Attack name	Type	Updating rule to derive x'_{i+1}
Full gradient	White-box	$x_i + \alpha_i \text{sign} \nabla_x \mathcal{L}\left((g_\phi \circ f_\theta)(x), y\right)\|_{x=x_i}$
Classifier PGD	Preprocessor-blind	$x_i + \alpha_i \text{sign} \nabla_x \mathcal{L}\left(g_\phi(x), y\right)\|_{x=x_i}$
BPDA (Athalye et al., 2018)	Adaptive	$x_i + \alpha_i \text{sign} \nabla_x \mathcal{L}\left(g_\phi(x), y\right)\|_{x=f_\theta(x_i)}$
Joint attack (score)	Adaptive	$x_i + \alpha_i \left(\varepsilon \text{sign}(s_\theta(x_i)) + (1-\varepsilon)\text{sign}(\nabla_x \mathcal{L}(g_\phi(x), y)\|_{x=x_i})\right)$
Joint attack (full)	Adaptive	$x_i + \alpha_i \left(\varepsilon \text{sign}(f_\theta(x_i) - x_i) + (1-\varepsilon)\text{sign}\nabla_x \mathcal{L}(g_\phi(x), y)\|_{x=x_i}\right)$
SPSA (Uesato et al., 2018)	Black-box	$x_i + \alpha_i \text{sign} \sum_{j=1}^N \frac{\mathcal{L}\left(\left((g_\phi \circ f_\theta)(x + \varepsilon v_j), y\right) - \mathcal{L}\left((g_\phi \circ f_\theta)(x - \varepsilon v_j), y\right)\right) \cdot v_j}{2N\varepsilon}$

Table 2. CIFAR-10 results for Preprocessor-blind attacks. The PGD attacks to the classifier is performed at ℓ_∞ ε-ball with $\varepsilon = 8/255$. The results borrowed from the references are marked with $*$.

Models	Accuracy (%) Standard	Robust	Architecture
Raw WideResNet	95.80	0.00	WRN-28-10
ADP ($\sigma = 0.1$)	93.09	85.45	WRN-28-10
ADP ($\sigma = 0.25$)	86.14	80.24	WRN-28-10
Adversarial purification methods			
(Hill et al., 2021)	84.12	78.91	WRN-28-10
(Shi et al., 2021)*	96.93	63.10	WRN-28-10
(Du & Mordatch, 2019)*	48.7	37.5	WRN-28-10
(Grathwohl et al., 2020)*	75.5	23.8	WRN-28-10
(Yang et al., 2019)*			
$\quad p = 0.8 \to 1.0$	94.9	82.5	ResNet-18
$\quad p = 0.6 \to 0.8$	92.1	80.3	ResNet-18
$\quad p = 0.4 \to 0.6$	89.2	77.4	ResNet-18
(Song et al., 2018)*			
$\quad$ Natural + PixelCNN	82	61	ResNet-62
$\quad$ AT + PixelCNN	90	70	ResNet-62
Adversarial training methods, transfer-based			
(Madry et al., 2018)*	87.3	70.2	ResNet-56
(Zhang et al., 2019)*	84.9	72.2	ResNet-56

Table 3. CIFAR-10 results for adaptive attacks at ℓ_∞ ε-ball with $\varepsilon = 8/255$. We compare our proposed method with other recently proposed preprocessor-based defense methods, and adversarial training methods with white-box attacks for reference. The results borrowed from the references are marked with $*$.

Models Attacks	Accuracy (%) Natural	Robust	Architecture
ADP ($\sigma = 0.25$)	86.14		
$\quad$ BPDA 40+EOT		**70.01**	WRN-28-10
$\quad$ BPDA 100+EOT		**69.71**	WRN-28-10
$\quad$ Joint (score)+EOT		70.61	WRN-28-10
$\quad$ Joint (full)+EOT		78.39	WRN-28-10
$\quad$ SPSA		80.80	WRN-28-10
ADP ($\sigma = 0.0$)	90.60		
$\quad$ BPDA		76.87	WRN-28-10
$\quad$ Joint (score)		80.81	WRN-28-10
$\quad$ Joint (full)		80.58	WRN-28-10
$\quad$ SPSA		47.6	WRN-28-10
Adversarial purification methods			
(Hill et al., 2021)			
$\quad$ BPDA 50+EOT	84.12	54.90	WRN-28-10
(Song et al., 2018)*			
$\quad$ BPDA	95.00	9	ResNet-62
(Yang et al., 2019)*			
$\quad$ BPDA 1000	94.8	40.8	ResNet-18
$\quad$ (+AT, $p = 0.4 \to 0.6$)	88.7	55.1	WRN-28-10
$\quad$ (+AT, $p = 0.6 \to 0.8$)	91.0	52.9	WRN-28-10
$\quad$ Approx. Input	89.4	41.5	ResNet-18
$\quad$ Approx. Input (+AT)	88.7	62.5	ResNet-18
(Shi et al., 2021)*			
$\quad$ Classifier PGD 20	91.89	53.58	WRN-28-10
Adversarial training methods			
(Madry et al., 2018)*	87.3	45.8	ResNet-18
(Zhang et al., 2019)*	84.90	56.43	ResNet-18
(Carmon et al., 2019)	89.67	63.1	WRN-28-10
(Gowal et al., 2020)*	89.48	64.08	WRN-28-10

the evaluation results of ADP for preprocessor-blind attacks on CIFAR-10 dataset, and compare them with other preprocessor-based methods and adversarial training methods. Except for (Shi et al., 2021), all the methods do not have any knowledge about the attacks during training.

Since the preprocessor-blind attacks are considered as a special case of *transfer-based black-box attacks*, we also compare ours to adversarial training methods tested with transfer-based black-box attacks (Madry et al., 2018; Zhang et al., 2019). The transfer-based black-box attacks assume that an attacker can access the training data, and thus can train a substitute model generating adversarial examples with them. The results for the transfer based attacks are borrowed from Dong et al. (2020).

We observe that ADP successfully purifies attacked images and shows high robust-accuracy on preprocessor-blind attacks while maintaining high natural accuracy.

5.1.2. EVALUATION: STRONG ADAPTIVE ATTACKS

We present our evaluation results for strong adaptive attacks on CIFAR-10 dataset as introduced in Section 5.1. Table 3 shows the evaluation results for various adaptive attacks. For BPDA and its variants, we assume that an attacker knows the exact step sizes used for the purification, and the attacks are designed with them. For SPSA attacks, we use 1,280 batch size to make the attack strong enough. For BPDA+EOT attack and its variants, we take 15 different noisy inputs and get expected loss over them for each attack step. We
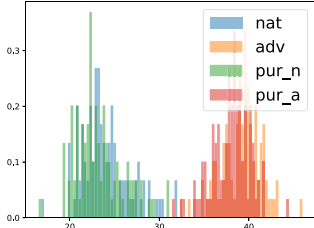
Figure 3. Histogram of score norms $\|s_\theta(x)\|_2$ for natural, adversarial, and purified images. Pur_a (pur_n) denotes score norms of one-step purified adversarial (natural) images. The x-axis and y-axis represent the score norm $\|s_\theta(x)\|_2$ and the density, respectively.
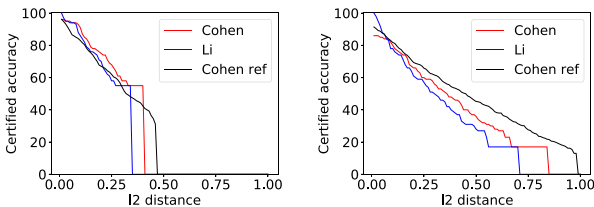

Figure 4. Approximate certified accuracy estimated by randomized smoothing. Left: $\sigma = 0.12$, Right: $\sigma = 0.25$, where σ is the noise level. The red and blue lines are the approximate certified accuracy of our model with respect to measures given from Cohen et al. (2019) and Li et al. (2018), respectively. The black line is the reference certified accuracy of randomized smoothing classifier from Cohen et al. (2019).

provide further results with different number of EOT attacks in Appendix D.3.

Ours generally shows the best results, successfully defending most of the strong adaptive attacks. One can see that, the random noise injection ($\sigma = 0.25$) generally improves the robust accuracy compared to the ones for without random noise injection ($\sigma = 0.0$). Especially, without random noise injection, ours fails on SPSA while it doesn't with random noise injection.

5.1.3. EVALUATION: CERTIFIED ROBUSTNESS

A classifier is *certifiably robust* within an area including an input x if the classifier predicts a constant label inside the area. A randomized smoothing classifier h_ϕ from the base classifier network g_ϕ is given by $h_\phi(x) = \mathrm{argmax}_{k \in [K]} [\mathbb{E}_\varepsilon[g_\phi(x + \varepsilon)]]_k$, where $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$. According to (Cohen et al., 2019), the certified radius of the randomized smoothing classifier is defined by

$$R = \frac{\sigma}{2}(\Phi^{-1}(p_A) - \Phi^{-1}(p_B)), \quad (19)$$

Table 4. CIFAR-10-C results for common corruption. The results borrowed from the references are marked with ∗.

	Accuracy	Noise	Blur	Weather	Digital
Raw WideResNet	71.89	42.37	72.74	86.23	78.83
ADP ($\sigma = 0.25$)	77.45	84.68	75.52	77.98	73.42
ADP ($\sigma = 0.25$)+Detection	78.96	84.57	71.76	84.34	76.60
ADP ($\sigma = 0.1$)	76.25	86.88	68.55	78.88	73.36
ADP, ($\sigma = 0.0$)	80.49	84.58	73.12	86.39	78.89
ADP, ($\sigma = 0.0$ + DCT)	80.74	84.96	73.11	87.26	78.69
ADP, ($\sigma = 0.0$ + AugMix)	82.63	87.52	75.20	88.82	80.20
ADP, ($\sigma = 0.0$ + DCT+AugMix)	82.40	85.05	75.80	88.11	81.30
Adversarial training methods					
(Zhang et al., 2019)	75.63	77.83	78.37	74.98	71.88
(Carmon et al., 2019)	80.40	81.20	83.44	80.19	76.98
(Cohen et al., 2019)	73.70	81.48	72.60	72.19	70.46
Training classifiers with augmentations					
(Hendrycks et al., 2020)*	88.78	84.66	89.68	90.93	88.47
(Wang et al., 2021)*	89.52	85.61	89.68	91.88	89.95
(Hossain et al., 2020)*	89.17	86.80	88.75	91.25	89.28

where p_A and p_B are the probabilities of top-2 probable labels after Gaussian noise $\mathcal{N}(0, \sigma^2 I)$ is injected, and Φ^{-1} denotes the inverse standard Gaussian CDF. We report our approximate certified accuracy in Fig. 4, by sampling noisy images for 100 times per image on 100 sampled images at the test set in CIFAR-10 dataset, and compare this to Cohen et al. (2019) evaluated for 500 sampled images. We show that we outperform previous results at noise level $\sigma = 0.12$, and show comparable certified accuracy at $\sigma = 0.25$.

5.1.4. EVALUATION: COMMON CORRUPTIONS

We present the evaluation results on CIFAR-10-C. The results are summarized in Table 4. Please refer to Appendix C for full results. We found that ADP with noise injection underperforms ADP without noise injection. Unlike the adversarial perturbations whose norms are bounded, the norms of the corruptions applied to images are not bounded, so they are not effectively screened by Gaussian noise injection. Still, the average performance with or without noise injection surpasses other adversarial training baselines (note that Carmon et al. (2019) uses extra unlabeled dataset so is not directly comparable). We could further improve the robust accuracy by exploiting Discrete Cosine Transform (DCT) and AugMix (Hendrycks et al., 2020) when training the score network. The idea is, instead of the typical Gaussian distribution, we can use the modified perturbation distribution $q(\tilde{x}|x) = \mathcal{N}(\tilde{x}|F(x), \sigma^2 I)$ where $F(x)$ is the augmentation obtained by either DCT or AugMix. Please refer to Appendix C for detailed description.

5.1.5. EVALUATION: EXTRA DATASETS

We also present our results for other image benchmarks, including MNIST, FashionMNIST, and CIFAR-100. For CIFAR-100, We provide additional results of extensive datasets in Appendix D.6. For CIFAR-100, ours achieved remarkable robust accuracy of **39.72%** against BPDA at-

tacks, outperforming the previous result (Li et al. (2020) (adversarial training method) reported the robust accuracy **28.88%**) by a large margin.

5.1.6. DETECTING ADVERSARIAL EXAMPLES

Finally, we show that we can detect adversarial examples using the norm of the score networks. Fig. 3 shows the difference of distributions of the score values for natural and adversarial images.

6. Conclusion

In this paper, we proposed a novel adversarial purification method with score-based generative models. We discovered that an EBM trained with DSM can quickly purify attacked images with deterministic short-run updates, and the purification process can further be robustified by injecting Gaussian noises before purification. We validated our method on various benchmark datasets using diverse types of adversarial attacks and demonstrated its superior performance.

Acknowledgement

References

Athalye, A., Carlini, N., and Wagner, D. A. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of The 35th International Conference on Machine Learning (ICML 2018)*, 2018.

Buckman, J., Roy, A., Raffel, C., and Goodfellow, I. J. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2018.

Carmon, Y., Raghunathan, A., Schmidt, L., Duchi, J. C., and Liang, P. S. Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2019.

Cohen, J. M., Rosenfeld, E., and Kolter, J. Z. Certified adversarial robustness via randomized smoothing. In *Proceedings of The 36th International Conference on Machine Learning (ICML 2019)*, 2019.

Croce, F., Andriushchenko, M., Sehwag, V., Flammarion, N., Chiang, M., Mittal, P., and Hein, M. Robustbench: a standardized adversarial robustness benchmark. *arXiv:2010.09670*, 2020.

Dhillon, G. S., Azizzadenesheli, K., Lipton, Z. C., Bernstein, J., Kossaifi, J., Khanna, A., and Anandkumar, A. Stochastic activation pruning for robust adversarial defense. In *International Conference on Learning Representations (ICLR)*, 2018.

Dong, Y., Fu, Q., Yang, X., Pang, T., Su, H., Xiao, Z., and Zhu, J. Benchmarking adversarial robustness on image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Du, Y. and Mordatch, I. Implicit generation and modeling with energy based models. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2019.

Garreau, D., Jitkrittum, W., and Kanagawa, M. Large sample analysis of the median heuristic. *arXiv:1707.07269*, 2018.

Ghosh, P., Losalka, A., and Black, M. J. Resisting adversarial attacks using gaussian mixture variational autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

Gowal, S., Qin, C., Uesato, J., Mann, T., and Kohli, P. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv:2010.03593*, 2020.

Grathwohl, W., Wang, K., Jacobsen, J., Duvenaud, D., Norouzi, M., and Swersky, K. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations (ICLR)*, 2020.

Guo, C., Rana, M., Cissé, M., and van der Maaten, L. Countering adversarial images using input transformations. In *International Conference on Learning Representations (ICLR)*, 2018.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of The 13th International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, 2010.

Hendrycks, D. and Dietterich, T. G. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*, 2019.

Hendrycks, D., Mu, N., Cubuk, E. D., Zoph, B., Gilmer, J., and Lakshminarayanan, B. Augmix: A simple data processing method to improve robustness and uncertainty. In *International Conference on Learning Representations (ICLR)*, 2020.

Hill, M., Mitchell, J. C., and Zhu, S.-C. Stochastic security: Adversarial defense using long-run dynamics of energy-based models. In *International Conference on Learning Representations (ICLR)*, 2021.

Hossain, M. T., Teng, S. W., Sohel, F., and Lu, G. Robust image classification using a low-pass activation function and dct augmentation. *arXiv:2007.09453*, 2020.

Hyvärinen, A. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 2005.

Kurakin, A., Goodfellow, I. J., and Bengio, S. Adversarial machine learning at scale. In *International Conference on Learning Representations (ICLR)*, 2017.

Lecun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. A tutorial on energy-based learning. *Predicting structured data*, 2006.

Li, B., Chen, C., Wang, W., and Carin, L. Second-order adversarial attack and certifiable robustness. *arXiv:1809.03113*, 2018.

Li, B., Wang, S., Jana, S., and Carin, L. Towards understanding fast adversarial training. *arXiv:2006.03089*, 2020.

Lin, G., Milan, A., Shen, C., and Reid, I. RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

Nijikamp, E., Zhu, S., and Wu, Y. N. Learning non-convergent non-persistent short-run MCMC toward energy-based model. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2019.

Pinot, R., Ettedgui, R., Rizk, G., Chevaleyre, Y., and Atif, J. Randomization matters how to defend against strong adversarial attacks. In *Proceedings of The 37th International Conference on Machine Learning (ICML 2020)*, 2020.

Salman, H., Li, J., Razenshteyn, I. P., Zhang, P., Zhang, H., Bubeck, S., and Yang, G. Provably robust deep learning via adversarially trained smoothed classifiers. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2019.

Samangouei, P., Kabkab, M., and Chellappa, R. Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations (ICLR)*, 2018.

Schott, L., Rauber, J., Bethge, M., and Brendel, W. Towards the first adversarially robust neural network model on MNIST. In *International Conference on Learning Representations (ICLR)*, 2019.

Shi, C., Holtz, C., and Mishne, G. Online adversarial purification based on self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2021.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2019.

Song, Y. and Ermon, S. Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.

Song, Y., Kim, T., Nowozin, S., Ermon, S., and Kushman, N. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2018.

Spall, J. C. A stochastic approximation technique for generating maximum likelihood parameter estimates. In *1987 American Control Conference*, 1987.

Srinivasan, V., Rohrer, C., Marbán, A., Müller, K., Samek, W., and Nakajima, S. Robustifying models against adversarial attacks by langevin dynamics. *Neural Networks*, 2021.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

Tramèr, F., Carlini, N., Brendel, W., and Madry, A. On adaptive attacks to adversarial example defenses. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.

Uesato, J., O'Donoghue, B., Kohli, P., and van den Oord, A. Adversarial risk and the dangers of evaluating against weak attacks. In *Proceedings of The 35th International Conference on Machine Learning (ICML 2018)*, 2018.

van der Maaten, L. and Hinton, G. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 2008.

Vincent, P. A connection between score matching and denoising autoencoders. *Neural Computation*, 2011.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P. Extracting and composing robust features with denoising autoencoders. In *Proceedings of The 25th International Conference on Machine Learning (ICML 2008)*, 2008.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 2010.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations (ICLR)*, 2021.

Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., and Gu, Q. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations (ICLR)*, 2020.

Xiao, C., Zhong, P., and Zheng, C. Enhancing adversarial defense by k-winners-take-all. In *International Conference on Learning Representations (ICLR)*, 2020.

Yang, Y., Zhang, G., Xu, Z., and Katabi, D. Me-net: Towards effective adversarial robustness with matrix estimation. In *Proceedings of The 36th International Conference on Machine Learning (ICML 2019)*, 2019.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *British Machine Vision Conference*, 2016.

Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. Theoretically principled trade-off between robustness and accuracy. In *Proceedings of The 36th International Conference on Machine Learning (ICML 2019)*, 2019.