Towards Understanding the Generative Capability of Adversarially Robust Classifiers

Yao Zhu[1]*, Jiacheng Ma[2], Jiacheng Sun[2]†, Zewei Chen[2], Rongxin Jiang[1], Yaowu Chen[1], Zhenguo Li[2]
[1]Zhejiang University, [2]Huawei Noah's Ark Lab

Abstract

Recently, some works found an interesting phenomenon that adversarially robust classifiers can generate good images comparable to generative models. We investigate this phenomenon from an energy perspective and provide a novel explanation. We reformulate adversarial example generation, adversarial training, and image generation in terms of an energy function. We find that adversarial training contributes to obtaining an energy function that is flat and has low energy around the real data, which is the key for generative capability. Based on our new understanding, we further propose a better adversarial training method, Joint Energy Adversarial Training (JEAT), which can generate high-quality images and achieve new state-of-the-art robustness under a wide range of attacks. The Inception Score of the images (CIFAR-10) generated by JEAT is 8.80, much better than original robust classifiers (7.50). In particular, we find that the robustness of JEAT is better than other hybrid models.

1. Introduction

Adversarial training can improve the robustness of a classifier against adversarial perturbations imperceptible to humans. Unlike a normal classifier, an adversarially robust classifier can generate good images by gradient descending the cross-entropy loss from random noises. Recently, some works discovered this phenomenon, and the quality of generated images is even comparable to GANs [21, 7]. The generative capability of an adversarially robust model from a classification task is interesting and surprising. However, it is unclear why an adversarially trained classifier can generate natural images. As image generation is a crucial topic, understanding the generative capability of adversarially robust classifiers could be inspiring and can give hints to many other generative methods.

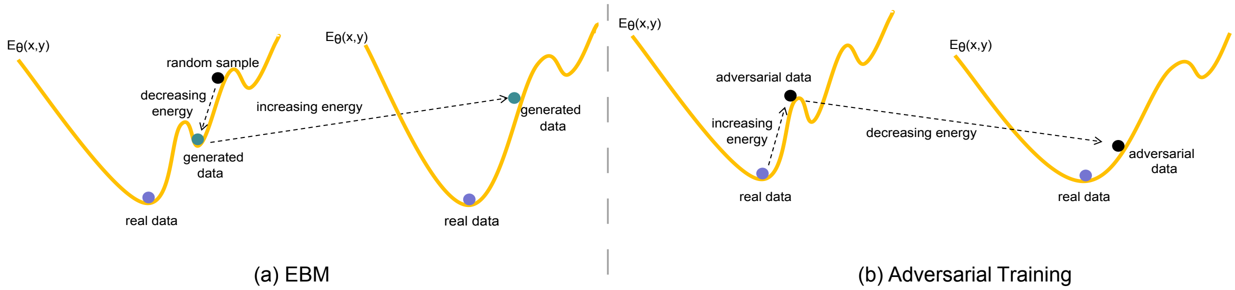
In this paper, we aim to understand the generative capability of an adversarially trained classifier and further improve the quality of generated images from the energy perspective. For Energy-Based Model (EBM) [10], it first generates low energy samples from random noises, then increases the energy of generated samples by updating model parameters. In this way, EBM can obtain a good energy function that is smooth and has low energy near the real data, as illustrates in Fig. 1(a). Thus, EBM can generate good images by sampling with Langevin Dynamics with the good energy function [5].

We show that adversarially trained classifiers can also obtain such a good energy function, which is flat and has low energy near the real data. This implies the generative capability of the adversarially robust classifier. For a classifier, we can define energy functions on the output logits. We reformulate the original adversarial training and image generation in terms of energy functions. In fact, the adversarial examples are high-energy samples near the real data, and adversarial training is trying to decrease the energy of these examples by updating model parameters. This procedure can also help us to learn a flat energy function with low energy near the real data, as illustrated in Fig. 1(b). Moreover, based on our understanding, we find another interesting phenomenon that a normal classifier is able to generate images if we add random noise to images during training. Though injecting noises to training data can generate high-energy samples, the perturbation direction may not be efficient compared with the adversarial attack. Thus, larger noise is needed in training for generating high-energy samples.

The cross-entropy loss can be expressed as the difference between $E_\theta(x, y)$ and $E_\theta(x)$. We show that $E_\theta(x, y)$, which is flat and has lower energy around the real data, plays a key role in the conditional generation task. However, making the energy $E_\theta(x, y)$ flat and low near the real data is just a by-product of original adversarial training. Thus we propose Joint Energy Adversarial Training (JEAT)

*This work was done when he was a research intern in Huawei Noah's Ark Lab. Email: ee_zhuy@zju.edu.cn.

†Corresponding to: Jiacheng Sun (sunjiacheng1@huawei.com)



(a) EBM

(b) Adversarial Training

Figure 1. EBM training and adversarial training make the energy functions smooth near the real data in different ways. (a) EBM starts from random samples, then moves along the direction of energy descent for fixed steps to obtain low energy samples. During optimization, the energy of these points is increased. Thus EBM obtains a smooth energy region around real data. (b) Adversarial examples are high-energy samples around real data. Adversarial training decreases the energy of adversarial examples by updating model parameters, thus obtains a flat region around real data.

to directly optimize the $E_\theta(x,y)$. We generate adversarial examples by directly increasing the $E_\theta(x,y)$ as Eq. (18) and update model parameters by maximizing the likelihood of joint distribution $p_\theta(x,y)$ as Eq. (19). We show that JEAT further improves the quality of the generated images.

The main contributions of our paper are summarized below:

- We propose a novel explanation of the image generation capability of a robust classifier from an energy perspective.
- We find the generative capability of normal classifiers with injecting noise in the training process and explain it from the energy point of view.
- We propose a training algorithm JEAT from an energy perspective that improves image generative capability.

2. Preliminaries

2.1. Adversarial Training

Adversarial training, first proposed in [8], can effectively defend against adversarial attacks by solving a bi-level min-max optimization problem [18]. It can be formulated as:

$$\begin{cases} \min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathcal{L}(x+\delta^*,y;\theta)], \\ \delta^* = \arg\max_{\|\delta\|_p\le\epsilon}\mathcal{L}(x+\delta,y;\theta), \end{cases} \tag{1}$$

where y is the ground-truth label of the input x, δ^* denotes the adversarial perturbation added to x, $\mathcal{L}$ denotes the loss function, $\|\cdot\|_p$ denotes the ℓ_p-norm that constrains the perturbated sample in an ℓ_p-ball with radius ϵ centered at x.

The ℓ_∞ adversarial perturbation is usually approximately solved by the fast gradient sign method (FGSM) [8] and projected gradient descent (PGD) [18]. For FGSM attacks δ_x takes the form:

$$\delta_x = \epsilon\ sign(\nabla_x\mathcal{L}(x,y;\theta)), \tag{2}$$

where $sign$ is the sign function. PGD attack is a kind of iterative variant of FGSM which generates an adversarial sample starting from a random position in the neighborhood of the clean image. PGD can be formulated as:

$$\begin{cases} x_n = x_{n-1} + \eta\cdot sign(\nabla_{x_{n-1}}\mathcal{L}(x_{n-1},y;\theta)), \\ x_n = clip(x_n, x_0-\epsilon, x_0+\epsilon), \end{cases} \tag{3}$$

where x_0 is a clean image and η is the perturbation step.

2.2. Image Generation of Robust Model

Generating images with an adversarially trained classifier is first raised by Santurkar, Madry, etc. [20] where they demonstrate that a robust classifier alone suffices to tackle various image synthesis tasks such as generation. For a given label y, image generation minimizes the loss $\mathcal{L}$ of label y by:

$$x' = x - \frac{\eta}{2}\cdot\nabla_x\mathcal{L}(x,y;\theta) + \sqrt{\eta}\epsilon. \tag{4}$$

Starting from sample $x_0 \sim \mathcal{N}(\mu_y,\Sigma_y)$ where $\mu_y = \mathbb{E}_{x\sim P_{x|y}}(x)$ and $\Sigma_y = \mathbb{E}_{x\sim P_{x|y}}((x-\mu_y)^T(x-\mu_y))$, we can obtain a better image by minimizing the loss $\mathcal{L}$. Some simple improvements such as choosing a more diverse distribution to start with could further improve the quality of generated images [20]. However, it is not the focus of this paper.

2.3. Energy Based Model

Energy Based Models [16, 10] show that any probability density $p(x)$ for x can be expressed as

$$p_\theta(x) = \frac{\exp(-E_\theta(x))}{Z_\theta}, \tag{5}$$

where $E_\theta(x)$ represents the energy of x and is modeled by neural network, $Z_\theta = \int\exp(-E_\theta(x))dx$ is the normalizing factor parameterized by θ also known as the partition function. The optimization of the energy-based model

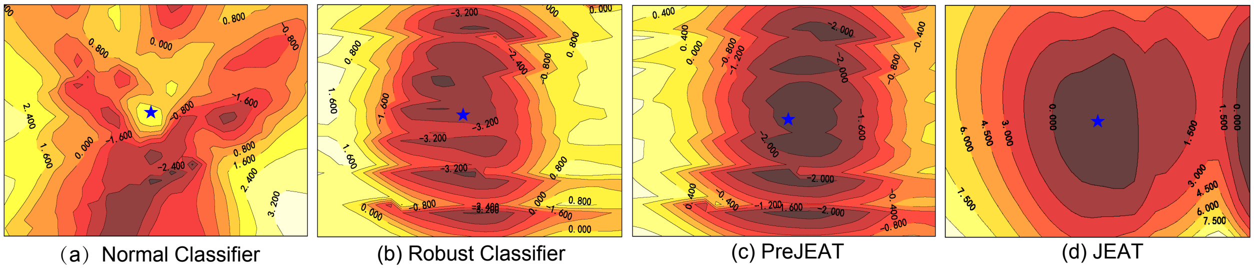

(a) Normal Classifier (b) Robust Classifier (c) PreJEAT (d) JEAT

Figure 2. Energy contour of different models. (a) The energy $E_\theta(x, y)$ contour of a naturally trained model; (b) The energy contour of an adversarially trained model; (c) The energy contour of PreJEAT (ours); (d) The energy contour of JEAT (ours). Darker colors in the plots represent lower energy values. The blue star point in the center of each figure is a chosen real image from CIFAR-10. We perturb the image in two random directions to get the energy landscape.

is through maximum likelihood learning by minimizing $\mathcal{L}_{ML}(\theta) = \mathbb{E}_{x\sim P_D}[-\log p_\theta(x)]$. The gradient is [5]:

$$\nabla_\theta \mathcal{L}_{ML}(\theta) = \mathbb{E}_{x^+\sim p_D}[\nabla_\theta E_\theta(x^+)] - \mathbb{E}_{x^-\sim p_\theta}[\nabla_\theta E_\theta(x^-)]. \tag{6}$$

Because sampling from p_θ is not feasible for an EBM, Langevin Dynamics is commonly used to approximately find samples from p_θ [11]. Thus, EBM training usually contains two stages: approximately generating samples from p_θ by Langevin Dynamics along the direction of energy descent and optimizing model parameters to increase the energy of these samples and decrease the energy of real samples by SGD. In this way, as illustrated in Fig. 1 (a), EBM could obtain a smooth energy function around real data and generate samples through Langevin Dynamics.

From energy perspective, we can also define $p_\theta(x, y)$ as follows:

$$p_\theta(x, y) = \frac{\exp(-E_\theta(x, y))}{\tilde{Z}_\theta}, \tag{7}$$

where $\tilde{Z}_\theta = \int \sum_y \exp(-E_\theta(x, y))dx$. Thus we also get $p_\theta(y|x)$ expressed by $E_\theta(x)$ and $E_\theta(x, y)$:

$$p_\theta(y|x) = \frac{p_\theta(x, y)}{p_\theta(x)} = \frac{\exp(-E_\theta(x, y)) \cdot Z_\theta}{\exp(-E_\theta(x)) \cdot \tilde{Z}_\theta}. \tag{8}$$

2.4. Sampling with Langevin Dynamics

Using $\nabla_x \log(p(x))$, Langevin Dynamics could generate samples from density distribution $p(x)$. The process starts from an initial point $\tilde{x}_0$ from a prior distribution π and recursively updates $\tilde{x}$ by:

$$\tilde{x}_t = \tilde{x}_{t-1} + \frac{\eta}{2}\nabla_{\tilde{x}_{t-1}} \log(p(\tilde{x}_{t-1})) + \sqrt{\eta}\epsilon, \tag{9}$$

where $\epsilon \sim \mathcal{N}(0, I)$ and η is a fixed step size. When $\eta \to 0$ and $T \to \infty$, $\tilde{x}_T$ is exactly an sample from $p(x)$ under some condition [31]. If we want to sample from $p_\theta(x, y)$ where y is a certain label, we could use Eq. (9) as well. Based on the energy framework as Eq. (7), we have $\nabla_x \log(p_\theta(x, y)) = -\nabla_x E_\theta(x, y)$. Hence the sampling process becomes:

$$\tilde{x}_t = \tilde{x}_{t-1} - \frac{\eta}{2}\nabla_{\tilde{x}_{t-1}} E_\theta(\tilde{x}_{t-1}, y) + \sqrt{\eta}\epsilon. \tag{10}$$

3. Energy Perspective on Robust Classifier

3.1. Energy Perspective on Adversarial Training

We denote $f(\cdot;\theta)$ as a classification neural network parameterized by θ. Let x be a sample. Then $f(x;\theta)[k]$ represents the k^{th} output of the last layer and we define $p_\theta(y|x)$ as:

$$p_\theta(y|x) = \frac{\exp(f(x;\theta)[y])}{\sum_{k=1}^{n}\exp(f(x;\theta)[k])}, \tag{11}$$

which resembles Boltzmann distribution, and n represents total possible classes. From Eq. (8) and (11), we define two energy functions as follows:

$$\begin{cases} E_\theta(x, y) = -\log(\exp(f(x;\theta)[y])), \\ E_\theta(x) = -\log(\sum_{k=1}^{n}\exp(f(x;\theta)[k])). \end{cases} \tag{12}$$

From Eq. (12), we have $\tilde{Z}_\theta = Z_\theta$. Furthermore, if the classification loss function is cross-entropy loss, it could also be expressed as:

$$\mathcal{L}(x, y;\theta) = E_\theta(x, y) - E_\theta(x). \tag{13}$$

In original adversarial training as in [18, 32, 24], by Fast Gradient Sign Method, adversarial example could be found by

$$x_{adv} = x + \eta \cdot sign(\nabla_x(E_\theta(x, y) - E_\theta(x))), \tag{14}$$

which is the direction of gradient ascent of loss defined in Eq. (13). The direction of adversarial perturbation relates to not only $E_\theta(x, y)$ but also $E_\theta(x)$. And in original adversarial training, the optimization process aims to decrease the loss in Eq. (13) by updating model parameters.

As we mentioned in Sec. 2.3, a good energy function $E_\theta(x, y)$ which is smooth and has low energy around real data is the key factor for generating good images for a given label y. We define the change of energy $E_\theta(x, y)$ after adversarial attack and after optimization as:

$$\begin{cases} \Delta_x E_\theta(x, y) = E_\theta(x_{adv}, y) - E_\theta(x_{ori}, y), \\ \Delta_\theta E_\theta(x, y) = E_{\theta_{updated}}(x_{adv}, y) - E_\theta(x_{adv}, y), \end{cases} \tag{15}$$

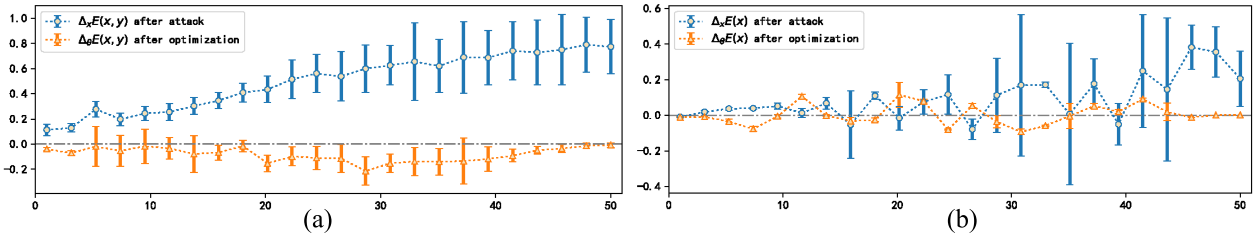


Figure 3. We illustrate the changes of energy in original adversarial training [18] on CIFAR-10 in 50 epochs (model has converged). The center points of the tags represent the mean value and the lengths represent the variance. (a) Adversarial examples increase the energy $E_\theta(x, y)$ as $\Delta_x E_\theta(x, y)>0$ during the training. The energy $E_\theta(x, y)$ of adversarial examples decrease after updating parameters as $\Delta_\theta E_\theta(x, y)<0$ during the training. (b) The value of $\Delta_\theta E_\theta(x)$ fluctuates around zero, sometimes positive and sometimes negative. Thus $E_\theta(x)$ has not been well optimized in the classification task.

where $E_\theta(x_{adv}, y)$ is the energy of x_{adv} given label y before updating parameters, $E_\theta(x_{ori}, y)$ is the energy of x_{ori} given label y before updating parameters, and $E_{\theta_{updated}}(x_{adv}, y)$ is the energy of x_{adv} given label y after updating parameters. The definition of $\Delta_x E_\theta(x)$ and $\Delta_\theta E_\theta(x)$ are similar in (15) with removing y.

As illustrated in Fig. 3 (a), adversarial attacks generate high-energy adversarial examples, and then the energy of adversarial examples is decreased by updating model parameters. Compared to EBM, adversarial training flattens the energy region around real data in a way different from EBM training. Adversarial training finds adversarial examples with high energy $E_\theta(x, y)$ near the data and then decreases the energy of those samples by updating model parameters. Noted that, in EBM, sampling from $p_\theta(x, y)$ by Langevin Dynamics follows the direction of energy descent, which is the opposite direction of adversarial examples. As illustrated in Fig. 1 (b), adversarial training can obtain a flat energy function around real data.

We illustrate energy $E_\theta(x, y)$ contours in Fig. 2 for a normal classifier and an adversarially trained classifier. From Fig. 2 (a), the energy function around the center is sharp for a normal classifier, and the energy of the real data is high. Because the low-energy region deviates from the center in a normal classifier, the generated images along the direction of energy descent are likely to fall into a region far away from the real data, which would not have good quality. For a robust classifier, the energy function near the center is smoother and lower, as shown in Fig. 2 (b).

Image generation by a robust classifier is introduced in Sec. 2.2, if we transform the loss function to energy expression as Eq. (13), the generating procedure of images becomes:

$$x' = x - \tfrac{\eta}{2} \cdot \nabla_x(E_\theta(x, y) - E_\theta(x)) + \sqrt{\eta}\epsilon, \qquad (16)$$

where $\epsilon \sim \mathcal{N}(0, I)$ and η is step size. The term $E_\theta(x, y) - E_\theta(x)$ in Eq. (16) implies that the generation iterations are related to both the energy of $E_\theta(x, y)$ and the energy of $E_\theta(x)$. By exploring low energy region of $E_\theta(x, y)$, which corresponds to high probability $p_\theta(x, y)$ region given constant normalizing factor from Eq. (7), we could obtain a good sample from high probability $p_\theta(x, y)$ region. Minimizing $E_\theta(x, y)$ contributes to maximizing $p_\theta(x, y)$ to generate images corresponding to the label y as shown in Eq. (10), but the energy of $E_\theta(x)$ is irrelevant to label y.

As illustrated in Fig. 3 (b), $E_\theta(x)$ has not been well optimized in the classification task, which may introduce label-independent noise. Thus, $E_\theta(x)$ may be a factor restricting the generative capability of the robust classifier, and we could drop $E_\theta(x)$ while only using $E_\theta(x, y)$ as:

$$x' = x - \frac{\eta}{2} \cdot \nabla_x(E_\theta(x, y)) + \sqrt{\eta}\epsilon. \qquad (17)$$

Eq. (17) is closely relate to sampling from Langevin Dynamics as Eq. (10). Approaching low energy region of $E_\theta(x, y)$ is equivalent to approaching the high probability region of $p_\theta(x, y)$. As shown in Fig. 4(a) and (b), using Eq. (17) gives better generated images than using Eq. (16).

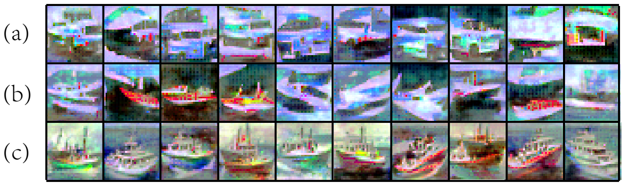
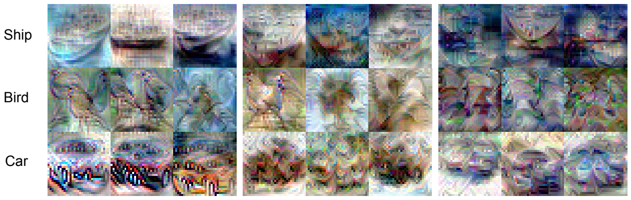


Figure 4. Given the label ship, the images generated by different methods. (a) shows ten images generated by original robust classifier using Eq. (16). (b) shows ten images generated by original robust classifier using Eq. (17). (c) shows ten images generated by robust classifier trained with adversarial examples got by Eq. (18) using Eq. (17).

3.2. Endowing Generative Capability to Normal Classifier

As analyzed before, we can generate high-energy samples from real data and reduce the energy of these samples during the optimization process to obtain a classifier with generative capability. The process of optimizing the energy $E_\theta(x, y)$ to be flat and low around real data contributes to generating good images. Adversarial training follows the



(a)Trained with big noise. (b)Trained with medium noise. (c)Trained with small noise.

Figure 5. The images generated by the models injected different strengths of noise in the training process. The clean accuracy of these models is almost the same. We use δ to denote the noise added to training data. (a) $\delta \sim \mathcal{N}(0, 24/255)$. Obviously, this model produces recognizable images. (b) $\delta \sim \mathcal{N}(0, 16/255)$, the shape of ship is visible. (c) $\delta \sim \mathcal{N}(0, 8/255)$, it is hard to recognize the shape of objects.

procedure and provides an effective way to find high-energy samples with small perturbations. Meanwhile, a simple random noise could also find high-energy samples but less effectively.

To verify our claim, we train normal classifiers by injecting different strengths of random noise into training data. As shown in Fig. 5, classifiers trained with different strengths of noise generate images of different qualities through the procedure in Sec. 3.1. Though the quality is not good enough, cars, birds and ships can be recognized with high confidence in Fig. 5 (a). From the comparison of images generated by adding different strengths of noise, we find that stronger random noise is needed to train a normal classifier in order to generate recognizable images. This is consistent with our explanation above that random noise is a "blind" adversarial direction to find high-energy samples. Because the direction is uncertain and very likely not the direction to efficiently increase energy, stronger perturbations may be needed. This experiment also shows that the process which generates high-energy samples from real data and reduces the energy of these samples during the optimization contributes to obtaining a classifier with generative capability.

3.3. Generating Adversarial Examples by Energy

As mentioned above, $E_\theta(x, y)$ determines the generative capability of a robust classifier. In original adversarial training, it does not increase the energy $E_\theta(x, y)$ of original data directly. In fact, only using $E_\theta(x, y)$ can also make x to be an adversarial sample x_{adv} as shown in Tab. 1.

Using $E_\theta(x, y)$ only in Eq. (14), we have:

$$x_{adv} = x + \eta \cdot sign(\nabla_x(E_\theta(x, y))). \tag{18}$$

Adversarial examples generated by energy attack in Eq. (18) have almost the same attack effect with the original normal attack by Eq. (14). The difference between x and x_{adv} is unrecognizable, but their energy are quite different. We name the method which trained with adversarial examples found by Eq. (18) and generates images by Eq. (17) after training as Preliminary Joint Energy Adversarial Training (PreJEAT).

Table 1. Normal attack and energy attack which just increases $E_\theta(x, y)$ on normal classifier (WideResNet-28-10). The perturbation radius is 8/255. They perform similarly on the CIFAR-10 and CIFAR-100 datasets and they can successfully attack the classifier.

Attack	CIFAR-10	CIFAR-100
No Attack	94.39%	78.54%
Normal Attack	0.12%	0.08%
Energy Attack	0.11%	0.10%

As we illustrate in Fig. 2, in a PreJEAT trained classifier, the energy around the natural image (center) is flatter and lower than a normal classifier and original robust classifier. In a robust classifier, the low energy region deviates from the center. But in the PreJEAT trained classifier, the natural image has the lowest energy. Thus, directly generating adversarial examples by $E_\theta(x, y)$ contributes to obtaining a flat and low energy function near the real data. By adversarial training with Eq. (18) we can obtain a robust classifier with better generative capability, as shown in Fig. 4(c).

3.4. Joint Energy Adversarial Training

We verified that a PreJEAT trained model has good generative capability in the previous section. However, there is still a discrepancy between training loss objective as Eq. (13) and adversarial training as Eq. (18) in PreJEAT. And we also find that the images generated by PreJEAT are not smooth enough in Fig. 4(c). We hope that the optimization process also optimizes $E_\theta(x, y)$ more directly to reduce the energy around real data. Hence we propose a new algorithm, Joint Energy Adversarial Training (JEAT), in Algorithm 1 to improve the generative capability of a classifier. In JEAT, we replace cross-entropy loss $-\log p_\theta(y|x)$ in PreJEAT with :

$$-\log p_\theta(x, y) = -\log p_\theta(y|x) - \log p_\theta(x), \tag{19}$$

where $p_\theta(x)$ is defined in Eq. (5). Optimizing $p_\theta(x, y)$ helps to get better $E_\theta(x, y)$ as shown in Eq. (7). The gradient of $\log p_\theta(x)$ is

$$\nabla_\theta \log(p_\theta(x)) = -\nabla_\theta E_\theta(x) - \mathbb{E}_{p_\theta(x)}[\nabla_\theta E_\theta(x)]. \tag{20}$$

We use Stochastic Gradient Langevin Dynamics (SGLD) to approximate $\mathbb{E}_{p_\theta(x)}$ [6].

JEAT uses Eq. (18) to find adversarial example and Eq. (17) to generate image like PreJEAT. With our proposed JEAT, adversarial examples, adversarial training, and image generation are connected to the energy $E_\theta(x, y)$ in a clear way. We also plot the energy contour of $E_\theta(x, y)$ in Fig. 2(d). Compared to the other three models, the energy function of a JEAT trained classifier near the center is the

Algorithm 1 Training and Generating of JEAT: Given network f, $E(x, y) = -\log(\exp(f(x)[y]))$ represent energy of (x, y), $E(x) = -\log(\sum_y \exp(f(x)[y]))$ represent energy of x, ℓ_∞ adversarial perturbation radius ϵ, SGLD step-size α, SGLD steps K, replay buffer B, reinitialization probility ρ, epochs T, dataset of size M, learning rate η.

```
Training:
for $i = 1, 2..., T$ do
    for $j = 1, 2..., M$ do
        ▶ Generating energy-based adversarial samples:
        $\delta = \mathcal{U}(-\epsilon, \epsilon)$
        $\delta = \delta + \epsilon \cdot sign(\nabla_x(E_\theta(x, y)))$
        $\delta = max(min(\delta, \epsilon), -\epsilon)$
        $x_{adv} = x_j + \delta$
        ▶ Generating samples by SGLD:
        $\tilde{x}_0 \sim \mathcal{U}(0, 1)$ with probability $\rho$, else $\tilde{x}_0 \sim B$
        for $t = 0, 1, 2, ..., K - 1$ do
            $\tilde{x}_{t+1} = \tilde{x}_t - \frac{\alpha}{2} \cdot \nabla_{x_t} E_\theta(\tilde{x}_t) + \sqrt{\alpha} \cdot \mathcal{N}(0, I)$
        end for
        Add $\tilde{x}_K$ to $B$
        ▶ Updating model parameters:
        $\nabla_\theta \mathcal{L}_{p(y|x_{adv})} = \nabla_\theta(E_\theta(x_{adv}, y) - E_\theta(x_{adv}))$
        $\nabla_\theta \mathcal{L}_{p(x_{adv})} = \nabla_\theta(E_\theta(x_{adv}) - E_\theta(\tilde{x}_K))$
        $\theta = \theta - \eta \cdot (\nabla_\theta \mathcal{L}_{p(y|x_{adv})} + \nabla_\theta \mathcal{L}_{p(x_{adv})})$
    end for
end for

Generating:
$x_0 \sim$ random sample
for $t = 0, 1, 2, ..., K - 1$ do
    $x_{t+1} = x_t - \frac{\alpha}{2} \cdot \nabla_{x_t} E_\theta(x_t, y) + \sqrt{\alpha} \cdot \mathcal{N}(0, I)$
end for
Output: $x_{gen} = x_K$
```

flattest among the four classifiers. Moreover, the energy contour is smooth across different energy levels. Hence images generated from a JEAT trained classifier are more likely a natural image and exhibits many natural details. Moreover, a flat energy function is less sensitive to noise perturbation. We will verify in experiments that JEAT improves both quality of generated images and adversarial robustness systematically.

4. Experiments

4.1. Experimental settings

To verify the validity of our energy-based explanation and the effectiveness of JEAT, we conducted experiments on CIFAR-10, CIFAR-100 and compare them with other methods. For network architecture, we use the WideResNet-28-10 for all algorithms. We repeat our experiments five times and report the mean and standard deviation.

In terms of image generation, we compare our generated pictures with JEM [10] and robust classifier [20]. We show the images generated by different algorithms and the scores of these methods based on Inception Score and Frechet Inception Distance metrics.

4.2. Image Generation

We use the procedure described in Sec. 3.1 to generate images by using different classifiers. The images generated by the adversarially trained classifier are shown in Fig. 6(a). As analyzed in Sec. 3.1, $E_\theta(x)$ can be the factor that restricts the generative capability of a robust model. We show the images generated by PreJEAT which uses $E_\theta(x, y)$ instead of $E_\theta(x, y) - E_\theta(x)$ both in adversarial training and generating images in Fig. 6(b). This approach greatly improves the quality of generated images. The energy plots of classifiers are presented in Sec. 3. By using PreJEAT, the energy is smoother and has a larger low energy region around real data, and it generates better quality images, especially in fine details and overall shape.

We also show the images generated by JEM [10] which is an energy-based model utilizing a classifier in Fig. 6(c). Compared to Fig. 6(b), JEM generates a more diverse background and is smooth in pixel. However, the features of objects stand out and are more perceptible to human in the images generated by PreJEAT.

As is shown in Fig. 6(d), our JEAT algorithm generates the best images compared to the other three. The reason behind is that JEAT is adversarially trained with energy $E_\theta(x, y)$ as in Eq. (18), and directly optimize $E_\theta(x, y)$ in the training contributes to getting better $E_\theta(x, y)$. The classifier trained by JEAT generates natural images with abundant features. Moreover, JEAT has the best performance based on the metrics of the Inception Score and Frechet Inception Distance.

We also validate the generalizability of the JEAT trained classifier in the sense that the images generated differ from any images in the training set, as shown in Fig. 7. Hence, JEAT is not just memorizing training images it has seen but generalizes as well.

4.3. Score Matching Metric

"Score" is defined here as $s_\theta(x, y) = \nabla_x \log p_\theta(x, y)$ [25]. Score matching tries to match the vector field of the gradient of $\log p_\theta(x, y)$ with respect to x to the real vector field given in data distribution [25] by minimizing Fisher Divergence:

$$\frac{1}{2} \cdot \mathbb{E}_{p_{data}(x|y)}[\|\nabla_x \log p_\theta(x, y) - \nabla_x \log p_{data}(x, y)\|_2^2]. \tag{21}$$

With the score defined in the distribution of (x, y) as $s_\theta(x, y)$, an equivalent optimization problem to (21) is min-

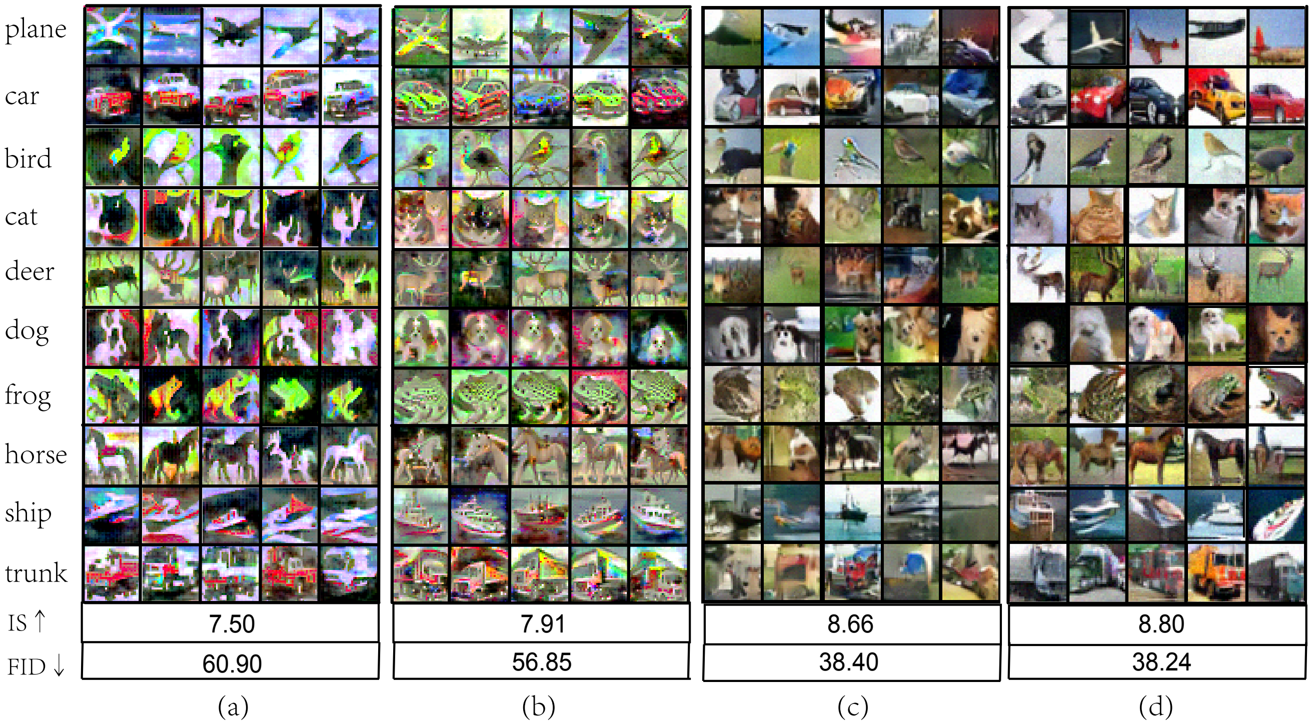


Figure 6. The generated images of different models. (a) The images generated by standard adversarially trained classifier; (b) The images generated by PreJEAT trained classifier; (c) The images generated by JEM [10] trained classifier; (d) The images generated by JEAT trained classifier. Images in each line are generated for a class in CIFAR-10.

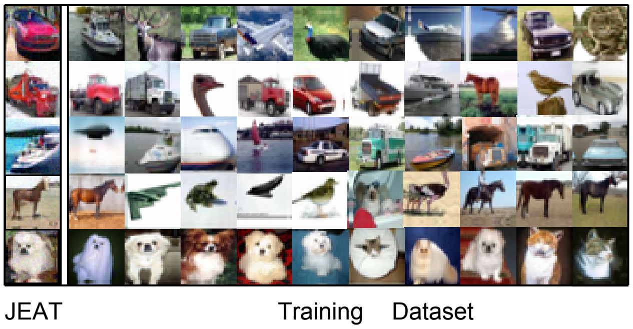


Figure 7. JEAT generates different images different from training data. The left column is generated by JEAT, and the right ten columns are images in the training dataset with the minimum distance and highest similarity to the image generated by JEAT. The image we generated is different from the original ones in the training dataset. We use SSIM [30] to measure similarity.

imizing

$$\mathbb{E}_{p_{data}(x|y)}[\tfrac{1}{2} \cdot \|s_\theta(x,y)\|_2^2 + tr(\nabla_x s_\theta(x,y))], \qquad (22)$$

where $\nabla_x s_\theta(x,y)$ denotes the Jacobian of $s_\theta(x,y)$ [13]. From energy perspective, score $s_\theta(x,y) = -\nabla_x E_\theta(x,y)$. Hence we denote objective in Eq. (22) as Joint Fisher Score (FS):

$$\mathbb{E}_{p_{data}(x|y)}[\tfrac{1}{2} \cdot \|\nabla_x E_\theta(x,y)\|_2^2 - tr(\nabla_x^2 E_\theta(x,y))]. \qquad (23)$$

It is a simple and effective metric that a lower value indicates $\nabla_x \log p_\theta(x,y)$ is closer to $\nabla_x \log p_{data}(x,y)$. So smaller value of Fisher Score corresponds to better generative capability (not considering failure cases here). Fisher Score could be used as a metric to assess the generative capability of models. Similarly, we denote Marginal Fisher Score (MFS) as:

$$\mathbb{E}_{p_{data}(x)}[\tfrac{1}{2} \cdot \|\nabla_x E_\theta(x)\|_2^2 - tr(\nabla_x^2 E_\theta(x))]. \qquad (24)$$

As shown in Tab. 2, both the Joint Fisher Score and the Marginal Fisher Score for our JEAT are the smallest in all the four models, which implies that the score of our model matches well with the ground truth distribution. This also means that JEAT can generate better images using Langevin Dynamics. The score matching metric is also consistent with the results of IS and FID, which are commonly used.

Table 2. Metrics for different models. "JFS" and "MFS" are the scores to evaluate the Joint and marginal Fisher Scores. "Normal" denotes normal training, "AT" denotes adversarial training.

Method	Normal	AT [20]	JEM [10]	JEAT
JFS↓	6.97	2.83	0.330	0.023
MFS↓	23.08	3.12	1.44	0.15
IS↑	-	7.50	8.66	8.80
FID↑	-	60.90	38.40	38.24

4.4. Robustness of JEAT

In this section, we show that the JEAT models not only can generate good images but also has comparable robustness with other hybrid models. We compare our model among hybrid models, including Glow [14], IGEBM [4], and JEM [10]. As shown in Tab. 3, we compare standard classification accuracy, and robust accuracy under ℓ_∞ PGD-20 attack with $\epsilon = 8/255$. The experimental results show that JEAT can both improve the generative capability and adversarial robustness of JEM.

JEM [10] is a well-written paper, which proposes that classifiers can also be trained as generative models and such models have the robustness comparable to adversarial training. However, when we use standard method [3] for evaluation, JEM's robustness is only 6.11% (under ℓ_∞ PGD-20, ϵ=8/255). We follow the same standard evaluation [3] method to test our JEAT's robustness and show it can improve from 6.11% to 30.55% (under ℓ_∞ PGD-20, ϵ=8/255).

Table 3. The performance comparison among four hybrid models on CIFAR-10. We use ℓ_∞ PGD-20 attack with $\epsilon = 8/255$.

Model	Standard Accuracy	Robustness
Glow [14]	67.6%	-
IGEBM [4]	45.06%	32.19%
JEM [10]	92.90%	6.11%[1]
JEAT	85.16%	30.55%

5. Related Work

5.1. Adversarial Training

Adversarial attacks can give the wrong prediction while adding negligible perturbations for humans to input data [28]. Adversarial training first proposed in [8] can effectively defend against such attacks by training on both clean data and adversarial examples. [18] formulates adversarial training as a bi-level min-max optimization problem and trains models exclusively on adversarial images rather than both clean and adversarial images. Although it effectively improves the adversarial robustness, expensive computational cost and performance degradation on clean images are the two fatal shortcomings of it. Many works try to reduce the computation cost to the natural training of a model. [24] updates the perturbation and weights at the same time, which reduces the bi-level problem to single-level optimization. [32] samples the perturbation randomly in every iteration. Many works try to mitigate the performance degradation on clean images. [34] gives a hint on how to balance the trade-off between vanilla and robust accuracy. [9] shows that additional unlabeled data can help to increase accuracy in adversarial training. Interestingly, robust optimization can be recast as a tool for enforcing priors on the features learned by deep neural networks [7]. Moreover, a robust classifier can tackle some challenging tasks in image synthesis [22].

5.2. Energy-based Model

Recently, the energy-based model has attracted significant attention. Effective estimating and sampling the partition function is the primary difficulty in training energy-based models [16, 12, 29]. Some research works have made a contribution to improve the training of energy-based models, such as sample the partition function through amortized generation [19, 2], utilize a separate generator network for negative image sample generations [15, 27] and score matching where the gradients of an energy function are trained to match the gradients of real data [26, 23]. Kevin Swersky et al. propose that treat classifier as an energy-based model can enable classifiers to generate samples rivaling the quality of recent GAN approaches [10]. Kyungmin et al. present a hybrid model which is built upon adversarial training and energy based training to deal with both out-of-distribution and adversarial examples [17].

Motivated by these discoveries, we investigate the generative capability of an adversarially trained model from an energy perspective and provide a novel explanation. We have further proposed an algorithm that can improve generation capacity.

6. Conclusion

We present a novel energy perspective on the generative capability of an adversarially trained classifier and propose our JEAT methods to obtain a classifier with stronger robustness that generates images with better quality. We find that a normal classifier can also generate images by injecting random noise in the training process, and we interpret this as blind adversarial training, for that random noise may find high energy examples accidentally. In summary, adversarial examples, blind or not, aim to find high-energy examples, and the classifier's optimization aims to lower the energy by optimizing model parameters. This process, as we validate, is beneficial for the robustness of the model and the quality of generated images. In addition, larger model capacity [9, 18], smoother activation function [9, 33] and unlabeled data [9, 1] are shown to improve adversarial robustness as well. We think these methods are promising ways to boost the performance of JEAT further and leave them for future work.

[1] We test the robustness of JEM's [10] open-source model (from https://github.com/wgrathwohl/JEM) by standard evaluation method [3] (https://robustbench.github.io/). [17] also shows that the robustness of JEM is 9.29% under the same setting (ℓ_∞ PGD-20, ϵ=8/255) by their implementation.

References

[1] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems*, pages 11192–11203, 2019.

[2] Bo Dai, Zhen Liu, Hanjun Dai, Niao He, Arthur Gretton, Le Song, and Dale Schuurmans. Exponential family estimation via adversarial dynamics embedding, 2020.

[3] Y. Dong, Q. Fu, X. Yang, T. Pang, H. Su, Z. Xiao, and J. Zhu. Benchmarking adversarial robustness on image classification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 318–328, Los Alamitos, CA, USA, jun 2020. IEEE Computer Society.

[4] Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *CoRR*, abs/1903.08689, 2019.

[5] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[6] Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models, 2020.

[7] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Adversarial robustness as a prior for learned representations, 2019.

[8] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *ICLR*, 2015.

[9] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020.

[10] Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one, 2020.

[11] Will Grathwohl, Kuan-Chieh Wang, Jorn-Henrik Jacobsen, David Duvenaud, and Richard Zemel. Learning the stein discrepancy for training and evaluating energy-based models without sampling, 2020.

[12] Matthew Hoffman, Pavel Sountsov, Joshua V. Dillon, Ian Langmore, Dustin Tran, and Srinivas Vasudevan. Neutralizing bad geometry in hamiltonian monte carlo using neural transport, 2019.

[13] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005.

[14] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in neural information processing systems*, pages 10215–10224, 2018.

[15] Rithesh Kumar, Sherjil Ozair, Anirudh Goyal, Aaron Courville, and Yoshua Bengio. Maximum entropy generators for energy-based models, 2019.

[16] Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu Jie Huang. A tutorial on energy-based learning, 2006.

[17] Kyungmin Lee, Hunmin Yang, and Se-Yoon Oh. Adversarial training on joint energy based model for robust classification and out-of-distribution detection. In *2020 20th International Conference on Control, Automation and Systems (ICCAS)*, pages 17–21, 2020.

[18] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[19] Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. Learning non-convergent non-persistent short-run mcmc toward energy-based model, 2019.

[20] Shibani Santurkar, Andrew Ilyas, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Image synthesis with a single (robust) classifier. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 1262–1273. Curran Associates, Inc., 2019.

[21] Shibani Santurkar, Dimitris Tsipras, Brandon Tran, Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Image synthesis with a single (robust) classifier, 2019.

[22] Shibani Santurkar, Dimitris Tsipras, Brandon Tran, Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Image synthesis with a single (robust) classifier, 2019.

[23] Saeed Saremi, Arash Mehrjou, Bernhard Schölkopf, and Aapo Hyvärinen. Deep energy estimator networks, 2018.

[24] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *Advances in Neural Information Processing Systems*, pages 3353–3364, 2019.

[25] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 11918–11930. Curran Associates, Inc., 2019.

[26] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution, 2020.

[27] Yunfu Song and Zhijian Ou. Generative modeling by inclusive neural random fields with applications in image generation and anomaly detection, 2020.

[28] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[29] Yulin Wang, Xuran Pan, Shiji Song, Hong Zhang, Gao Huang, and Cheng Wu. Implicit semantic data augmentation for deep networks. In *Advances in Neural Information Processing Systems*, pages 12614–12623, 2019.

[30] Zhou Wang, Member, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity, 2004.

[31] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference*

on Machine Learning, ICML'11, page 681–688, Madison, WI, USA, 2011. Omnipress.

[32] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.

[33] Cihang Xie, Mingxing Tan, Boqing Gong, Alan Yuille, and Quoc V Le. Smooth adversarial training. *arXiv preprint arXiv:2006.14536*, 2020.

[34] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy, 2019.