Exploring the Connection between Robust and Generative Models

Senad Beadini[1], Iacopo Masi[1,*]

[1] Sapienza, University of Rome, Computer Science Department

Abstract

We offer a study that connects robust discriminative classifiers trained with adversarial training (AT) with generative modeling in the form of Energy-based Models (EBM). We do so by decomposing the loss of a discriminative classifier and showing that the discriminative model is also aware of the input data density. Though a common assumption is that adversarial points leave the manifold of the input data, our study finds out that, surprisingly, untargeted adversarial points in the input space are very likely under the generative model hidden inside the discriminative classifier—have low energy in the EBM. We present two evidence: untargeted attacks are even more likely than the natural data and their likelihood increases as the attack strength increases. This allows us to easily detect them and craft a novel attack called High-Energy PGD that fools the classifier yet has energy similar to the data set. The code is available at github.com/senad96/Robust-Generative

Keywords

robust models, generative AI, adversarial attacks

1. Introduction

Although the attention to machine learning robustness and its security implications became suddenly of interest, the idea of *robust* of machine learning (ML) is not new. Pioneering articles back in 2004 raised awareness that the community naively assumed that the data generation process in data mining is independent of the system responses [1]. A few years later, [2] provided the first consolidated perspective of adversarial ML and the need for a change in how ML is thought, conceived, and applied. Standard, non-robust ML has three limits in the adversarial setting: (1) the first limitation is more proper of the generalization ability of ML systems. These models operate under the assumption that the testing data is sampled from the same unknown generation process that created the data–i.i.d. assumption, independent and identically distributed. This is a too strict assumption when actual input data can be sampled from another generation process–o.o.d., out of distribution; (2) more importantly, the generation process is assumed to be disconnected from the system capabilities of taking decisions (3) minimization of the empirical risk does not guarantee to be resilient to an adversary. To address those issues, the formalization of minimizing the maximum risk [3] by A. Wald of 1944 came at hand to cast the empirical risk minimization into *adversarial risk minimization* [4, 5], following a proper definition.

In this paper, following the interpretation in [6] of a discriminative classifier $p_{\boldsymbol{\theta}}(y|\mathbf{x})$ implemented by a Convolutional Deep Neural Network (DNN) for multi-class classification, we offer a new view on adversarial perturbations connecting them with generative modeling. Following [6, 7], we analyze a robust model under the lens of Energy-based Models (EBM), and we decompose the cross-entropy loss of $p_{\boldsymbol{\theta}}(y|\mathbf{x})$ to show that it contains the notion of the joint distribution of the data and the labels $p_{\boldsymbol{\theta}}(\mathbf{x}, y)$ along with the distribution of the data itself $p_{\boldsymbol{\theta}}(\mathbf{x})$. Although adversarial perturbations are known as input points that switch the decision boundary—thus affecting $p_{\boldsymbol{\theta}}(y|\mathbf{x})$—we show that there is a strong dependency also with $p_{\boldsymbol{\theta}}(\mathbf{x})$. In the scenario of untargeted attacks, we show that an increase in the attack strength implies a significant increase of $p_{\boldsymbol{\theta}}(\mathbf{x})$. The increase is

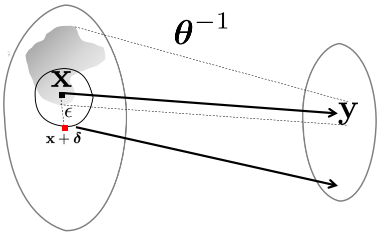

Figure 1: An input point $\mathbf{x}$ on the left is mapped to a latent code $\mathbf{y}$ to solve a generic task (i.e. classification). When we invert the latent code $\mathbf{y}$ and we go back in the input space, we have a full set of solutions. Adversarial examples are points *not* in that set and also very close to $\mathbf{x}$ yet mapping to another latent code, far from $\mathbf{y}$: for small variations of the input, we have large variations of the output. Adversarial training fixes these points, making the model more “aware” of the distribution of the input data.

Ital-IA 2023: 3rd National Conference on Artificial Intelligence, organized by CINI, May 29–31, 2023, Pisa, Italy

*Corresponding author.

beadini.1754617@studenti.uniroma1.it (S. Beadini); masi@di.uniroma1.it (I. Masi)

0000-0003-0444-7646 (I. Masi)

CEUR Workshop Proceedings (CEUR-WS.org)

so substantial that the attacks' energy surpasses the magnitude of the energy of natural data. Under the lens of generative modeling via EBM, we hope to shed some light on a few phenomena that emerge from robust classifiers, such as: 1) robust models are naturally better calibrated than standard classifiers, 2) robust models offer gradients of the loss wrt to the input with more structure, and better "explain" the data 3) the trade-off between robust and natural accuracy. Fig. 1 shows the idea of training a robust model: in the standard case, when we train a discriminative classifier, we model $p_{\boldsymbol{\theta}}(y|\mathbf{x})$, and thus we seek for a function $\boldsymbol{\theta}$ that maps the fixed input point $\mathbf{x}$ to a latent code $\mathbf{y}$ that is useful for the task, for instance, minimize the cross-entropy loss. A discriminative model ignores the input data distribution, thereby adjusting its parameters to separate the data. Furthermore, the input space is usually larger than the embedding space, and the model has to build invariance to nuances of the training data to achieve classification. What we *do not enforce*, instead, is that the inverse of $\boldsymbol{\theta}$ has to fall within a boundary of the input data. In other words, when we train the discriminative model, the function is not bijective, and inverting the function gives as output a set. Adversarial samples are points in the input space very close to $\mathbf{x}$ yet map to a different latent code far from $\mathbf{y}$. Adversarial Training (AT) [4, 5] restores the "right" connection.

In this paper we make the following contributions:

◇ Viewing a discriminative model under the lens of an EBM, we show that the more we increase the strength of an untargeted attack, the more $p_{\boldsymbol{\theta}}(\mathbf{x})$ increases, even more than the points naturally present in the dataset.

◇ We show that the energy of an input point $\mathbf{x}$ can be used as a strong detector for adversarial data. The more the attack increases its strength, the easier to detect. We do so by performing *no training* of the detector except choosing a threshold.

◇ We offer a way to bypass the aforementioned detector demonstrating that is possible to design an adversarial perturbation that crosses the decision boundary yet its energy is kept similar to those of natural data.

◇ Finally, we emphasize that a robust or quasi-robust model can be effectively employed to enhance image synthesis in generative AI, supported by a recent paper by [8], where image synthesis is performed by inverting a quasi-robust model.

2. Prior Work

Robust and Generative Models. We believe the first connection between robust and generative models has been done in the paper by Grathwoh *et al.* [6], where for the first time, a discriminative DNN ending with the softmax classifier is interpreted as an EBM. [6] proposes JEM to train the DNN in a hybrid way—to be both discriminative and generative. They also show that hybrid training somehow slightly improves robustness to adversarial attacks. JEM was extended in [9] to stabilize and speed up the training. Later on, [7] established the first connection on how adversarial training (AT) and EBM bend the energy function in a different way yet using a similar contrastive methodology. Very recently [10] moved in the same direction of demystifying the generative capabilities of robust models developing a unified probabilistic framework, and working also in unsupervised settings. Finally, [11] showed that AT improves Joint Energy-Based Generative Modelling, while incorporating into the modeling a sharpness-aware minimization (SAM) procedure [12].

Synthesis with a robust model. At the best of our knowledge [13] is the first paper to employ a *robust* classifier for synthesis since it is shown that robust models attain input gradients that better correlated with "human perception" [14, 15]. The community argued in [16] that AT renders the discriminative model invertible, while [17] believes that AT yields gradients closer to the image manifold, which is in line with the connection between robust and generative model. The generative capabilities of robust models have been used by [18] for solving inverse problems and for controllable image synthesis [8].

3. Preliminaries

In Section 3.1, we will briefly review the settings of adversarial attacks in a white box scenario, while Section 3.2 we examine data density modeling of the input data using an Energy-based Model (EBM).

3.1. Adversarial Perturbations

A DNN $\boldsymbol{\theta} : \mathbb{R}^X \rightarrow \mathbb{R}^Y$ is a universal approximator for generic functions [19] that maps an input data $\mathbf{x}$ into another meaningful code $\mathbf{y}$ using a loss function $\mathcal{L}$. In machine vision, usually X is high dimensional, with $X \gg Y$, and $\mathcal{L}$ guides $\boldsymbol{\theta}$ in solving the task at hand. The definition of adversary specifies that, though $\boldsymbol{\theta}$ may perform well on the task yet $\boldsymbol{\theta}$ is brittle since is possible to obtain a slightly perturbed instance of $\mathbf{x}^\star$ that fools $\boldsymbol{\theta}$ on the same task such that $\mathbf{x}^\star \approx \mathbf{x}$. The gist of all the attacks follows an inverse optimization process in which the attacker optimizes an adversarial point $\mathbf{x}^\star \in \mathbb{R}^X$ in the input space so that the model ascends the loss in the output space (i.e. untargeted attack) or makes a confident response of an erroneous label, decided a priori by the attacker (i.e. targeted attack). The attacker often yields a constraint on how much information needs to be changed in the input. In most of the current attacks, the bound on the information is implemented with pixel-wise L_p norm as $\boldsymbol{\delta} \doteq \mathbf{x}^\star - \mathbf{x}$ and $||\boldsymbol{\delta}||_p \leq \epsilon$. All the attacks follow the above procedure with some differences on how many steps are taken to ascend the loss: one for Fast-Gradient Sign Method (FGSM) with its variants [4, 20], multi-steps with projections for Projected Gradient Descent (PGD) [5] or in how the loss is designed, e.g. Carlini & Wagner (CW) attack.

Attacks can be crafted in targeted or untargeted settings even in the physical world [21] and can transfer between different models. Whilst attack literature is solid, defenses have several limitations [22]: despite being computationally expensive, the only defense to withstand is Adversarial Training (AT) [4, 5]. Though in this paper we deal mainly with the image domain, adversarial perturbations are known in other domains (Natural Language Processing and tabular data) as counterfactual explanations [23]. Below we briefly review the two main methods to generate perturbations of the input given a DNN.

Fast Gradient Sign Method. FGSM [4] is a L_∞-norm adversarial attack and it is described in the following formula below:

$$\mathbf{x}^* = \mathbf{x} + \epsilon \ \text{sign}\Big(\ \nabla_{\mathbf{x}}\big[\mathcal{L}(\boldsymbol{\theta}(\mathbf{x}),\ y)\big]\ \Big) \tag{1}$$

where sign returns the sign of its argument.

Projected Gradient Descent. PGD can be seen as an iterative generalization of the FGSM algorithm [5]. The iterative procedure is shown below:

$$\mathbf{x}^* = \text{clip}_\epsilon\ \Big(\mathbf{x}^* + \alpha\ \text{sign}\Big(\ \nabla_{\mathbf{x}^*}\mathcal{L}(\boldsymbol{\theta}(\mathbf{x}^*),\ y)\ \Big)\Big) \tag{2}$$

where clip denotes the function that projects its argument to the surface of $\mathbf{x}$'s neighbor ϵ-ball while α is the step size. This algorithm is able to produce strong adversarial data that have a high probability to fool classifiers and commonly is used as a benchmark. Additional variants of the PGD attack, such as APGD [24], are accessible and can improve its reliability.

3.2. Energy-Based Model (EBM)

EBMs [25] are based on the idea that any probability density $p_{\boldsymbol\theta}(\mathbf{x})$ for $\mathbf{x} \in \mathbb{R}^X$ can be expressed via a Boltzmann distribution as:

$$p_{\boldsymbol\theta}(\mathbf{x}) = \frac{\exp\left(-E_{\boldsymbol\theta}(\mathbf{x})\right)}{Z(\boldsymbol\theta)} \tag{3}$$

where $E_{\boldsymbol\theta}(\mathbf{x})$ is called energy, modeled as a neural network, that maps each input $\mathbf{x}$ to a scalar. $Z(\boldsymbol\theta)$ is the normalizing constant, known as the partition function, such that $p_{\boldsymbol\theta}(\mathbf{x})$ is a valid probability density function; the challenge for training EBMs is approximating the constant $Z(\boldsymbol\theta)$. Training an EBM is performed via maximum likelihood estimation by minimizing the negative log-likelihood of the data:

$$\mathcal{L}_{ML}(\boldsymbol\theta) = \mathbb{E}_{\mathbf{x}\sim P_D}\Big[-\log p_{\boldsymbol\theta}(\mathbf{x})\Big] \tag{4}$$

Nevertheless, the latter is not effortless and several sampling methods have been designed to approximate it efficiently [26]. Specifically, it is known that the derivative of Eq. (4) w.r.t $\boldsymbol\theta$ is:

$$\nabla_{\boldsymbol\theta}\mathcal{L}_{ML}(\boldsymbol\theta) = \mathbb{E}_{\mathbf{x}^+\sim P_D}\Big[\nabla_{\boldsymbol\theta}E_{\boldsymbol\theta}(\mathbf{x}^+)\Big] - \mathbb{E}_{\mathbf{x}^-\sim p_{\boldsymbol\theta}}\Big[\nabla_{\boldsymbol\theta}E_{\boldsymbol\theta}(\mathbf{x}^-)\Big] \tag{5}$$

Unfortunately, for EBMs we can not sample from $p_{\boldsymbol\theta}$, therefore several MCMC methods have been proposed to sample from that density function [26]. For instance, in [6] Stochastic Gradient Langevin Dynamics (SGLD) has been used to train EBMs using gradient knowledge. SGLD sample with the following procedure:

$$\mathbf{x}_0 \sim p_0(\mathbf{x}) \quad , \quad \mathbf{x}_{t+1} = \mathbf{x}_t - \frac{\alpha}{2}\frac{\partial E_{\boldsymbol\theta}(\mathbf{x})}{\partial\boldsymbol\theta} + \alpha\epsilon \tag{6}$$

where $p_0(\mathbf{x})$ is typically a Uniform distribution and $\epsilon \sim \mathcal{N}(0, 1)$. Thus, we can summarize that the training of EBMs consists of two steps: generating approximate data samples from $p_{\boldsymbol\theta}$ using MCMC methods and optimizing the model parameters to increase the energy of these samples and decreasing the energy of the real samples via gradient descent.

DNN as an EBM. Recently, Grathwohl *et al.* [6] made a connection between DNN classifiers and EBMs. They show that an EBM is hidden inside a standard classifier and we can re-interpret the logits of a model $\boldsymbol\theta$ to build an EBM to compute $p(\mathbf{x}, y)$ and $p(\mathbf{x})$:

$$p_{\boldsymbol\theta}(\mathbf{x}, y) \ = \ \frac{\exp\left(\boldsymbol\theta(\mathbf{x})\ [y]\right)}{Z(\boldsymbol\theta)} \tag{7}$$

where $Z(\boldsymbol\theta)$ is the unknown normalizing constant, $\boldsymbol\theta(\mathbf{x})[k]$ is the k-th index of logits $\boldsymbol\theta(\mathbf{x})$, and $E_{\boldsymbol\theta}(\mathbf{x}, y) = -\boldsymbol\theta(\mathbf{x})[y]$ is the energy function. Following [6] we can compute $p_{\boldsymbol\theta}(y\ |\mathbf{x})$ and $p_{\boldsymbol\theta}(\mathbf{x})$ as:

$$p_{\boldsymbol\theta}(y|\mathbf{x}) = \frac{p_{\boldsymbol\theta}(\mathbf{x}, y)}{p_{\boldsymbol\theta}(\mathbf{x})} = \frac{\exp\big(-E_{\boldsymbol\theta}(\mathbf{x}, y)\big)}{\sum_{k=1}^{K}\exp\big(\boldsymbol\theta(\mathbf{x})[k]\big)}$$

$$p_{\boldsymbol\theta}(\mathbf{x}) = \sum_{k=1}^{K} p_{\boldsymbol\theta}(\mathbf{x}, y = k) = \frac{\sum_{k=1}^{K}\exp\left(\boldsymbol\theta(\mathbf{x})[k]\right)}{Z(\boldsymbol\theta)}$$

with K the number of classes. Notice that, for any classifier $\boldsymbol\theta$, the energy of a data point $\mathbf{x}$ is:

$$E_{\boldsymbol\theta}(\mathbf{x}) \ = \ -\log\sum_{k=1}^{K}\exp\big(\boldsymbol\theta(\mathbf{x})[k]\big) \tag{8}$$

Indeed a data point with a high probability of occurrence—high $p_{\boldsymbol\theta}(\mathbf{x})$—is equivalent to having low energy.

4. Method

4.1. Untargeted attacks have low $E_{\boldsymbol\theta}(\mathbf{x})$

Analyzing the energy $E_{\boldsymbol\theta}(\mathbf{x})$ gives us the opportunity to examine and study the generative model inside classifiers.

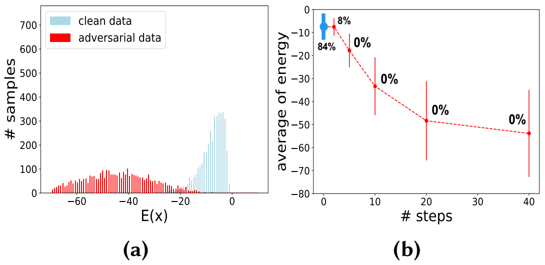

Figure 2: (a) Energy distribution $E_{\boldsymbol{\theta}}(\mathbf{x})$ for PGD-based adversarial and natural data on the test set of imagenette, given a standard non-robust DNN. Notice how the distributions are almost separated. **(b)** Each point indicates the mean absolute value of the energy and its spread taken at different PGD "steps" (increasing strength). Above each point we reported the accuracy. 🟥 indicates adv. while 🟦 natural data.

To better understand the characteristics of adversarial examples, we performed an energy analysis of DNNs for both adversarial examples generated using the untargeted PGD attack and natural data. Our results indicate a substantial difference in energy between adversarial and natural data, with adversarial points having much lower energy than natural points. These findings are exhibited in Fig. 2, which displays the energy for two different models on diverse datasets. In Fig. 2(a), the #steps (iterations) used is 40 in PGD and ϵ is always 8/255. Indeed, our analysis reveals an intriguing dependency between the energy function $E_{\boldsymbol{\theta}}(\mathbf{x})$ and the "strength" of an untargeted adversarial attack. It appears that the energy of adversarial data decreases as the strength of the attack increases. In our case, we define the attack strength as the discrete value of the number of steps of a PGD attack. Considering these results and reinterpreting them using the EBM framework, it can be inferred the following. Beyond the classic idea that adversarial attacks cross the decision boundary, we show that a DNN tends to "believe" that adversarial data are highly likely under the hidden generative model $p_{\boldsymbol{\theta}}(\mathbf{x})$. More surprisingly, they are even more likely than the natural data itself.

4.2. Detecting Adversarial Noise with the Energy Function

Based on the previous observations we propose the construction of a very simple detection algorithm for catching adversarial data using the energy function as a 1D discriminant. This idea has already been applied in [27] to detect out-of-distribution (OOD) data; however, there is a key difference from [27] and us: while the OOD data tend to move to higher values of $E_{\boldsymbol{\theta}}(\mathbf{x})$, in our case, we have the opposite; additionally, to the best of our knowledge, this technique was not yet applied to adversarial example detection. Our strategy detects PGD adversarial perturbations at inference time in a DNN equipped with a softmax classifier, without any additional computation overhead and any additional parameters. Firstly, the approach estimates a threshold t in the domain of $E_{\boldsymbol{\theta}}(\mathbf{x})$ using a validation set that balances the true positive rate (TPR) and false negative rate (FNR) based on a chosen metric. Subsequently, it uses this threshold to perform detection:

$$G(\boldsymbol{\theta}, \mathbf{x}) = \begin{cases} 1 & \text{if } \; -\log \sum_{k=1}^{K} \exp\big(\boldsymbol{\theta}(\mathbf{x})[k]\big) \leq t \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

4.3. Attacking The Energy Detector

The development of strong adversarial detectors is a significant research area, as such detectors typically depend on specific types of attacks. Although the energy approach is effective for identifying adversarial data with low energy, such as those generated via PGD, a crucial question is whether it is possible to generate perturbations that can mislead the aforementioned method. That means whether it is possible to produce adversarial examples that own energy comparable to natural data.

High-Energy PGD. In order to generate adversarial examples with similar energy as natural data, we propose to introduce a regularization term during the optimization step to force the creation of perturbations that lead to adversarial data with higher energy. The regularizer will remove the energy gap between natural and adversarial data. Specifically, we define the following optimization problem:

$$\arg\max_{\boldsymbol{\delta}} \Big[\mathcal{L}\big(\; \boldsymbol{\theta}(\mathbf{x}+\boldsymbol{\delta}), y\big) + \lambda \; E_{\boldsymbol{\theta}}(\mathbf{x}+\boldsymbol{\delta})\Big] \tag{10}$$

where $\mathcal{L}$ is the loss function, like cross-entropy and λ is a hyper-parameter that controls the strength of the regularization. Eq. (11) describes the High-Energy PGD attack (HE-PGD) implemented as:

$$\mathbf{x}^* = \text{clip}_\epsilon \left[\mathbf{x}^* + \alpha \, \text{sign}\left[\nabla_{\mathbf{x}^*} \mathcal{L}\big(\boldsymbol{\theta}(\mathbf{x}^*), y\big) + \lambda E_{\boldsymbol{\theta}}(\mathbf{x}^*)\right]\right] \tag{11}$$

where the initial $\mathbf{x}_0^*$ = $\mathbf{x} + \boldsymbol{\delta}$ with $\boldsymbol{\delta} \sim$ Uniform(-ϵ, ϵ). The choice of λ is crucial and can significantly affect optimization and thus the aggressiveness of the attack. The empirical value of the λ parameter is described in Section 5.

5. Experimental Results

In this section, we describe our experiments and confirm the efficacy of the energy function as a detection method for PGD-based attacks. Then, we investigate a connection between robust and generative models.

5.1. Detection Results

We use the imagenette [28], CIFAR-10 [29], and CIFAR-100 [29] datasets as benchmarks. All attacks except HE-PGD are implemented using torchattacks [30]. HE-PGD is implemented directly in PyTorch with a λ parameter of 1.2.

Evaluation Metrics. We measure the following metrics: 1) detection rate (DR); 2) false positive rate (FPR). We utilize the G-mean metric to estimate the threshold; G-mean is defined as the squared root of the product of sensitivity and specificity. Table 1 shows the detector's performance on diverse attack settings. Energy as a detector achieves larger DR for PGD-based perturbations on CIFAR-10, also scale to imagenette [28] where the input space is much larger than CIFAR, and works for unseen attacks such as APGD [24].

Dataset	Defense	Attack	DR	FPR
imagenette [28]	**Energy** (ResNet10)	PGD (8)	98.24	1.37
		PGD (16)	99.6	0.00
CIFAR-10 [29]	**Energy** (ResNet10)	PGD (8)	98.38	1.62
		APGD (8)	85.45	1.19
	KD+BU[31]	PGD (8)	92.27	0.96
	LID [32]	PGD (8)	94.39	1.81

Table 1
Attack detection with energy function. KD+BU and LID results taken from [33]. All numbers are percentages. Attacks are expressed with the (ϵ) budget.

5.2. HE-PGD Results

Our experiments in Fig. 3 show that HE-PGD is at least as effective as conventional PGD while it keeps the energy of the adversarial points similar to the natural data. Indeed in Fig. 3(a) the two distributions now largely overlap since the adversarial samples generated by HE-PGD have higher energy than those produced by PGD and are better aligned with the energy of the natural data. Consequently, when the value of λ is chosen appropriately, HE-PGD can bypass the energy-based detector, presented in Section 4.2. The stepsize of all attacks is fixed at 1/255.

From Fig. 3(b), we note that, unlike before, $E_{\boldsymbol{\theta}}(\mathbf{x})$ is approximately invariant to the number of steps (#steps) in projected gradient descent. At the same time, the classifier accuracy still drops to zero for higher iterations yet it does so slightly more gradually than before.

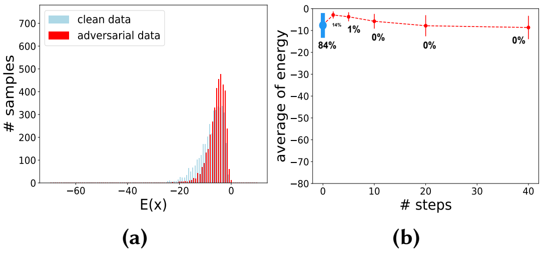

Figure 3: **(a)** On imagenette, energy distribution for HE-PGD adversarial and natural data. The two distributions are overlapped, which makes the energy detector worthless. **(b)** HE-PGD enforces invariance of $E_{\boldsymbol{\theta}}(\mathbf{x})$ wrt to the #steps yet still pushes the accuracy to 0% after a few steps. 🟥 indicates adv. while 🟦 natural data.

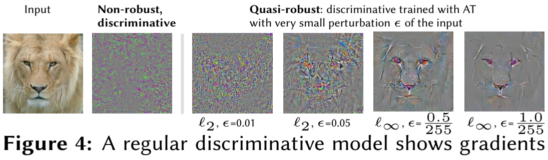
Input
Non-robust, discriminative
Quasi-robust: discriminative trained with AT with very small perturbation ϵ of the input
ℓ2, ϵ=0.01
ℓ2, ϵ=0.05
ℓ∞, ϵ=0.5/255
ℓ∞, ϵ=1.0/255

Figure 4: A regular discriminative model shows gradients of the loss respect to the input with sparse activations compared to robust models while these latter are more meaningful which better motivate them for synthesis and connect them to generative models.

5.3. Synthesis with a Robust Model

The evidence that robust models have input gradients aligned with semantic properties of the object, as shown in Fig. 4, can be explained under the lens of generative modeling. Aggregating gradients of the loss over the input can be seen as a way to sample points from the generative model hidden inside the discriminative classifier. What AT may be doing is rendering it a hybrid model that becomes generative by estimating gradients of the data distribution as a score-matching model [34]. There is also a strong parallelism between adversarial attacks and sampling from score-matching using SGLD—Eq. (6). Recently [8] showed that is possible to invert a quasi-robust model to perform convincing image synthesis. A quasi-robust model is a model which is non-robust from the point of view of security yet it inherits generative capabilities and keeps its well-performing accuracy. To generate the samples, [8] takes steps in the direction of the gradients that better carries information of class conditional data distribution $p(\mathbf{x}|y)$.

6. Conclusions

In this paper, by viewing a robust discriminative model as an EBM, we were able to detect adversarial example effectively and demonstrate that there is the possibility of bypassing the detector introducing a new attacked called High-Energy PGD. We believe that exploring the connection between robust and generative modeling may shed some light on interesting properties of robust classifiers.

Acknowledgment. The authors thank Dr. Mozhdeh Rouhsedaghat for providing the figure of the gradient of the standard and quasi-robust models. This project is partially supported by Sapienza research project "Prebunking: predicting and mitigating coordinated inauthentic behaviors in social media" and PNRR PE01 FAIR project.

References

[1] N. Dalvi, P. Domingos, S. Sanghai, D. Verma, Adversarial classification, in: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 2004, pp. 99–108.

[2] M. Barreno, B. Nelson, A. D. Joseph, J. D. Tygar, The security of machine learning, Machine Learning 81 (2010) 121–148.

[3] A. Wald, Statistical decision functions which minimize the maximum risk, Annals of Mathematics (1945) 265–280.

[4] I. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in: ICLR, 2015.

[5] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in: ICLR, 2018.

[6] W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, M. Norouzi, K. Swersky, Your classifier is secretly an energy based model and you should treat it like one, in: ICLR, 2020.

[7] Y. Zhu, J. Ma, J. Sun, Z. Chen, R. Jiang, Y. Chen, Z. Li, Towards understanding the generative capability of adversarially robust classifiers, in: ICCV, 2021, pp. 7728–7737.

[8] M. Rouhsedaghat, M. Monajatipoor, C.-C. J. Kuo, I. Masi, MAGIC: Mask-guided image synthesis by inverting a quasi-robust classifier, in: AAAI Conference on Artificial Intelligence, 2023.

[9] X. Yang, S. Ji, Jem++: Improved techniques for training jem, in: ICCV, 2021, pp. 6494–6503.

[10] Y. Wang, Y. Wang, J. Yang, Z. Lin, A unified contrastive energy-based model for understanding the generative ability of adversarial training, in: International Conference on Learning Representations, 2022.

[11] R. Korst, A. Asadulaev, Adversarial training improves joint energy-based generative modelling, arXiv:2207.08950 (2022).

[12] P. Foret, A. Kleiner, H. Mobahi, B. Neyshabur, Sharpness-aware minimization for efficiently improving generalization, in: ICLR, 2021.

[13] S. Santurkar, D. Tsipras, B. Tran, A. Ilyas, L. Engstrom, A. Madry, Image synthesis with a single (robust) classifier, in: NeurIPS, 2019.

[14] G. Aggarwal, A. Sinha, N. Kumari, M. Singh, On the benefits of models with perceptually-aligned gradients, in: ICLR, 2020.

[15] S. Kaur, J. Cohen, Z. C. Lipton, Are perceptually-aligned gradients a general property of robust classifiers?, arXiv:1910.08640 (2019).

[16] M. Terzi, A. Achille, M. Maggipinto, G. A. Susto, Adversarial training reduces information and improves transferability, arXiv:2007.11259 (2020).

[17] B. Kim, J. Seo, T. Jeon, Bridging adversarial robustness and gradient interpretability, in: ICLR Workshops, 2019.

[18] R. A. Rojas-Gomez, R. A. Yeh, M. N. Do, A. Nguyen, Inverting adversarially robust networks for image synthesis, arXiv:2106.06927 (2021).

[19] B. C. Csáji, et al., Approximation with artificial neural networks, Faculty of Sciences, Etvs Lornd University, Hungary 24 (2001) 7.

[20] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, J. Li, Boosting adversarial attacks with momentum, arXiv preprint (2018).

[21] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial machine learning at scale, in: ICLR, 2017.

[22] F. Tramer, N. Carlini, W. Brendel, A. Madry, On adaptive attacks to adversarial example defenses, arXiv:2002.08347 (2020).

[23] N. Madaan, I. Padhi, N. Panwar, D. Saha, Generate your counterfactuals: Towards controlled counterfactual generation for text, in: AAAI Conference on Artificial Intelligence, volume 35, 2021, pp. 13516–13524.

[24] F. Croce, M. Hein, Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, in: International conference on machine learning, PMLR, 2020, pp. 2206–2216.

[25] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, F. Huang, A tutorial on energy-based learning, Predicting structured data 1 (2006).

[26] Y. Song, D. P. Kingma, How to train your energy-based models, arXiv:2101.03288 (2021).

[27] W. Liu, X. Wang, J. Owens, Y. Li, Energy-based out-of-distribution detection, Advances in neural information processing systems 33 (2020) 21464–21475.

[28] fastai, imagenette, 2023. URL: https://github.com/fastai/imagenette.

[29] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images, Technical Report, Citeseer, 2009.

[30] H. Kim, Torchattacks: A pytorch repository for adversarial attacks, arXiv:2010.01950 (2020).

[31] R. Feinman, R. R. Curtin, S. Shintre, A. B. Gardner, Detecting adversarial samples from artifacts, arXiv:1703.00410 (2017).

[32] X. Ma, B. Li, Y. Wang, S. M. Erfani, S. Wijewickrema, G. Schoenebeck, D. Song, M. E. Houle, J. Bailey, Characterizing adversarial subspaces using local intrinsic dimensionality, arXiv:1801.02613 (2018).

[33] A. Aldahdooh, W. Hamidouche, S. A. Fezza, O. Déforges, Adversarial example detection for dnn models: A review and experimental comparison, Artificial Intelligence Review 55 (2022) 4403–4462.

[34] Y. Song, S. Ermon, Generative modeling by estimating gradients of the data distribution, in: NeurIPS, 2019.