Are Generative Classifiers More Robust to Adversarial Attacks?

Bruno Amorim de Araujo, Franck Laborde, Killian Steunou

Abstract

This report presents our work on the article Are Generative Classifiers More Robust to Adversarial Attacks? [9]. We implemented the authors experiment on MNIST [4], and applied the methods on the German Traffic Sign Recognition Benchmark dataset [13], under black-box adversarial attacks, and were unable to conclude on whether generative classifiers were more robust to adversarial attacks than discriminative classifiers.

The code used for our experiment is available at https://github.com/killian31/DeepBayesTorch.

1. Introduction

In recent years, the field of machine learning has seen significant advancements, particularly in the development of deep neural networks (DNNs). However, these advances have also exposed critical vulnerabilities, notably in their susceptibility to adversarial attacks [6, 14]. Adversarial examples, often imperceptible modifications to input data [1, 3, 6, 8, 10, 12], can lead to incorrect classifications by neural networks, raising concerns about their reliability in critical applications such as autonomous vehicles and medical diagnosis.

Generative classifiers [11], which model the joint probability distribution $p(x, y)$ of inputs x and labels y, have been proposed as a potentially more robust alternative to discriminative classifiers, which model $p(y|x)$. Deep Bayes classifiers, an improvement on classical Naive Bayes models, leverage deep latent variable models (LVMs) trained via variational inference. These models offer promising features, such as the ability to reject out-of-distribution inputs by estimating their likelihood under the data distribution.

In this report, we first verify the original article's results, and explore the robustness of generative classifiers under black-box adversarial attacks. Specifically, we focus on applying these methods to the German Traffic Sign Recognition Benchmark (GTSRB) dataset [13], a domain where robustness is extremely important. Traffic sign recognition systems are vital for autonomous driving, but are particularly susceptible to real-world adversarial attacks, such as strategically placed stickers that alter the classification outcome without raising suspicion.

We replicated the experiments from the article *Are Generative Classifiers More Robust to Adversarial Attacks?*[9] and extended them to the traffic sign dataset under black-box attack settings, with two attacks: *Gaussian*, and *Sticker*, that will be explained in section 5. Our aim was to simulate scenarios that could occur in real life, analyzing whether the robustness of generative classifiers could contribute to the development of more reliable critical systems, such as those in autonomous vehicles. By studying the interaction between adversarial attacks and detection mechanisms, we seek insights that may guide the design of secure machine learning frameworks.

This report begins with details of the data and methods used, followed by a detailed explanation of the generative classifiers and detection strategies. The experimental results are then presented, highlighting the challenges and findings, culminating in a discussion on the implications for improving robustness in critical systems.

2. Datasets

In this section, we describe the datasets used in our experiments. We focus on reproducing results on the MNIST dataset [4] and then extend our experiments to the GTSRB dataset [13].

2.1. MNIST

The MNIST dataset is a benchmark for handwritten digit classification. It contains 60,000 training and 10,000 testing grayscale images of size 28×28, representing digits from 0 to 9. For our experiments, pixel values were normalized to the range $[0, 1]$. Figure 1a shows examples.

2.2. GTSRB

The GTSRB dataset is a real-world classification challenge with 43 classes of traffic signs and over 50,000 images. These images are diverse and reflect varying conditions, such as lighting and perspective, making it ideal for evaluating the robustness of classifiers. To prepare the dataset, we resized all images to the size 32×32. Pixel values were then normalized to the range $[0, 1]$. Figure 1b shows examples.

(a) MNIST examples.

(b) GTSRB examples.

Figure 1. Examples of datasets used in the experiments. Left: handwritten digits from MNIST. Right: traffic signs from GTSRB.

3. Deep Bayes: conditional deep LVM as a generative classifier

In this section, we describe the seven models used in our experiments, focusing on their architectures, underlying mathematical formulations, and their application as generative and discriminative classifiers. We also detail the encoder-decoder and generator architectures employed for MNIST and GTSRB datasets.

3.1. Latent Variable Models (LVMs)

LVMs introduce unobserved latent variables z to model the joint distribution $p(x, y)$ of inputs x and labels y. The joint distribution is expressed as:

$$p(x, z, y) = p(z)p(y|z)p(x|z, y),$$

where $p(z)$ is the prior distribution over the latent variables, $p(y|z)$ models the dependency of the labels on the latent variables, and $p(x|z, y)$ represents the conditional distribution of the input given the latent variables and the labels.

Deep generative models, such as Variational Autoencoders (VAEs) [7], approximate $p(x|z, y)$ using neural networks. The training involves optimizing the variational lower bound:

$$\mathbb{E}_{\mathcal{D}}[\mathcal{L}_{VI}(x, y)] = \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_q\left[\log\frac{p(x_n, z_n, y_n)}{q(z_n|x_n, y_n)}\right],$$

where $\mathcal{D} = \{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$ is the training data and $q(z|x, y)$ is the amortized approximate posterior.
After training, the generative classifiers predict the label y^* for a given input x^* by approximating Bayes' rule using importance sampling. Specifically, the predicted class probability is computed as:

$$p(y^*|x^*) \approx \text{softmax}_{c=1}^{C}\left[\log\frac{1}{K}\sum_{k=1}^{K}\frac{p(x^*, z_c^k, y_c)}{q(z_c^k|x^*, y_c)}\right],$$

where $z_c^k \sim q(z|x^*, y_c)$ are samples from the variational posterior, $p(x^*, z_c^k, y_c)$ is the joint distribution, and $q(z_c^k|x^*, y_c)$ is the approximate posterior.

3.2. The Models

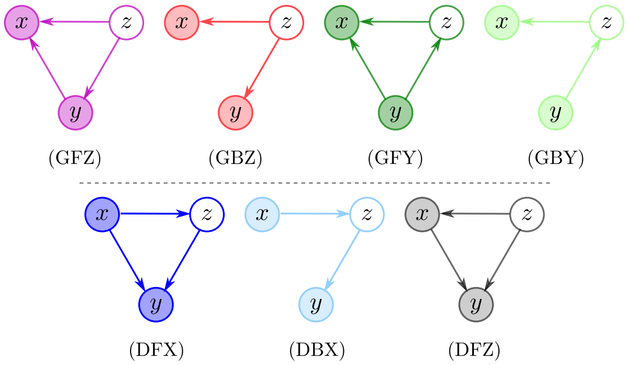

Figure 2. A visualisation of the graphical models, including both Generative (top row) and Discriminative ones (bottom row), as well as Fully connected and Bottleneck ones. The last character indicates the first node in the topological order of the graph.

We evaluated seven models, four generative and three discriminative classifiers, as defined in [9]. Each model follows a distinct factorization of $p(x, z, y)$ (see Figure 2). The naming convention of the models (e.g., GFZ, GBY, etc.) indicates whether the model is Generative (G) or Discriminative (D), Fully connected (F) or Bottleneck (B), and the first node in the probabilistic graph's topological order: the label (Y), the latent variable (Z), or the input (X).

$$p(x, z, y) = p(z)p(y|z)p(x|z, y) \quad \text{(GFZ)}$$
$$p(x, z, y) = p_{\mathcal{D}}(y)p(z|y)p(x|z, y) \quad \text{(GFY)}$$
$$p(x, z, y) = p(z)p(y|z)p(x|z) \quad \text{(GBZ)}$$
$$p(x, z, y) = p(y)p(z|y)p(x|z) \quad \text{(GBY)}$$
$$p(x, z, y) = p(x)p(z|x)p(y|z, x) \quad \text{(DFX)}$$
$$p(x, z, y) = p(z)p(x|z)p(y|z, x) \quad \text{(DFZ)}$$
$$p(x, z, y) = p(x)p(z|x)p(y|z) \quad \text{(DBX)}$$

3.3. Generative vs. Discriminative Properties

Generative classifiers model $p(x, y)$, allowing them to reject inputs far from the data manifold by estimating $p(x)$ or $p(x, y)$. Discriminative classifiers, which directly estimate $p(y|x)$, are optimized for classification but lack the ability to assess input likelihood. The different factorizations of $p(x, z, y)$ enable diverse robustness and detection capabilities, as explored in our experiments. We follow [9]'s guidelines and code for training the models.

4. Detecting adversarial attacks with generative classifiers

In this section, we describe the three detection methods used in our experience. In fact, the paper gives three detection methods for adversarial example using the generative classifier's logit values. The aim is to improve the robustness of the model by detecting whether the input image can be trusted or not. If an image is incorrectly labeled, for example, an image $\boldsymbol{x}_{cat}$ is labeled as a "dog", either the image is ambiguous and we cannot do anything at this level, or for a well-trained generative classifier, the logit $\log(\boldsymbol{x}_{cat}, \text{"dog"})$ will be low. This make it possible for us to detect adversarial images using the logit $\log(\boldsymbol{x}, y_c)$, with $c \in \{1, \ldots, C\}$ the class and $\boldsymbol{x}$ a test input. The purpose of the detection algorithm is then to reject both unlabeled inputs $\boldsymbol{x}$ that have low probability under $p(\boldsymbol{x})$ and labeled data $(\boldsymbol{x}, \boldsymbol{y})$ that have low probability $p(\boldsymbol{x}, \boldsymbol{y})$.

4.1. Marginal detection

The aim of this algortithm is to reject the input data that are far from the data manifold.

To do so, we need to select a threshold δ such that we reject an input $\boldsymbol{x}$ if $-\log p(\boldsymbol{x}) > \delta$. To find a good threshold, we can compute $\bar{\mu}_{\mathcal{D}} = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[-\log p(\boldsymbol{x})]$ and $\bar{\sigma}_{\mathcal{D}} = \sqrt{\mathbb{V}_{\boldsymbol{x}\sim\mathcal{D}}[\log p(\boldsymbol{x})]}$ with $\mathcal{D}$ the distribution for the inputs data. We can then compute $\delta = \bar{\mu}_{\mathcal{D}} + \alpha\bar{\sigma}_{\mathcal{D}}$ with α a parameter choosen to exclude a certain percentage of the data.

4.2. Logit detection

For this algorithm, the goal is to reject the data that are far from the joint density.

For a given attacked model $\boldsymbol{y} = F(\boldsymbol{x})$, one can reject $\boldsymbol{x}$ if $-\log p(\boldsymbol{x}, F(\boldsymbol{x})) > \delta_{\boldsymbol{y}}$. As in the previous section, we can use the mean and the variance $\bar{\mu}_c$ and $\bar{\sigma}_c$ computed on $p(\boldsymbol{x}, \boldsymbol{y}_c)$ for all class $c = 1, \ldots, C$ and set $\delta_{\boldsymbol{y}_c} = \bar{\mu}_c + \alpha\bar{\sigma}_c$ with α having the same role as previously.

4.3. Divergence detection

This algorithm aims at rejecting inputs with over-confident and/or under-confident predictions.

Let $p(\boldsymbol{x})$ represent a C-dimensional probability vector produced by the classifier. For each class c, we first compute the mean classification probability vector $p_c = \mathbb{E}_{(\boldsymbol{x},y_c)\in\mathcal{D}}[p(\boldsymbol{x})]$. Next, we calculate the mean μ_c and standard deviation σ_c for a chosen divergence or distance measure $D[p_c \| p(\boldsymbol{x})]$ across all samples $(\boldsymbol{x}, y_c) \in \mathcal{D}$. For a test input $\boldsymbol{x}^*$ with predicted label $c^* = \arg\max p(\boldsymbol{x}^*)$, it is rejected if $D[p_{c^*} \| p(\boldsymbol{x}^*)] > \mu_{c^*} + \alpha\sigma_{c^*}$.

Thus, an example $\boldsymbol{x}^*$ is rejected if its probability vector $p(\boldsymbol{x}^*)$ deviates significantly from those observed during training.

For the experiments performed in the section 5, we have use the KL-divergence as a criterion.

5. Experiments

5.1. Attacks

5.1.1 White-box Attacks

The white-box attack is characterized by the attacker having access to the model's architecture, parameters and training data. The implementations of the attacks followed the same structure as in [2], performed under the classifier, where that attacker could differentiate through the classifier, but unaware of the detector.

l_∞ **attacks**

- Fast Gradient Sign Method (FGSM):
  This method computes the gradient of the loss functions with respect to the input. Then it adds a perturbation, scaled by ε in the direction of the gradient's sign [6]:

$$x_{adv} = x + \varepsilon \cdot sign(\nabla_x L(x, y_{true}))$$

- Projected Gradient Descent (PGD):
  The PGD attack is build on the Fast Gradient Sign Method (FGSM) and is an iterative approach. At each iteration the loss is computed and the adversarial input is updated with regard to the step in direction of the gradient [10]. Each adversarial attack also need to be constrained by $||x_{adv} - x||_\infty \leq \varepsilon$.

$$x_{t+1} = x_t + \alpha \cdot sign(\nabla_{x_t} L(x_t, y_{true}))$$

- Momentum Iterative Method (MIM):
  This method introduces the concept of momentum, by using velocity vector g_t that accumulates gradients across T iterations, weighted by a momentum factor μ. The method continues to use the maximization of the loss function, constrained by the L_∞ norm. The adversarial example is updated using the sign of the velocity vector, and the perturbation is projected back into the valid input space to ensure it meets the specified constraints [5].

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_x L(f(x'_t), y)}{\|\nabla_x L(f(x'_t), y)\|_1}$$

$$x'_{t+1} = x'_t + \frac{\varepsilon}{T} \cdot \text{sign}(g_{t+1})$$

l_2 **attacks**

- Carlini & Wagner method (CW):
  This method differentiate itself from the others by formulating the adversarial example generation as an optimization problem. The method relies on the L_2 norm to constrain the adversarial noise added to the input, but also

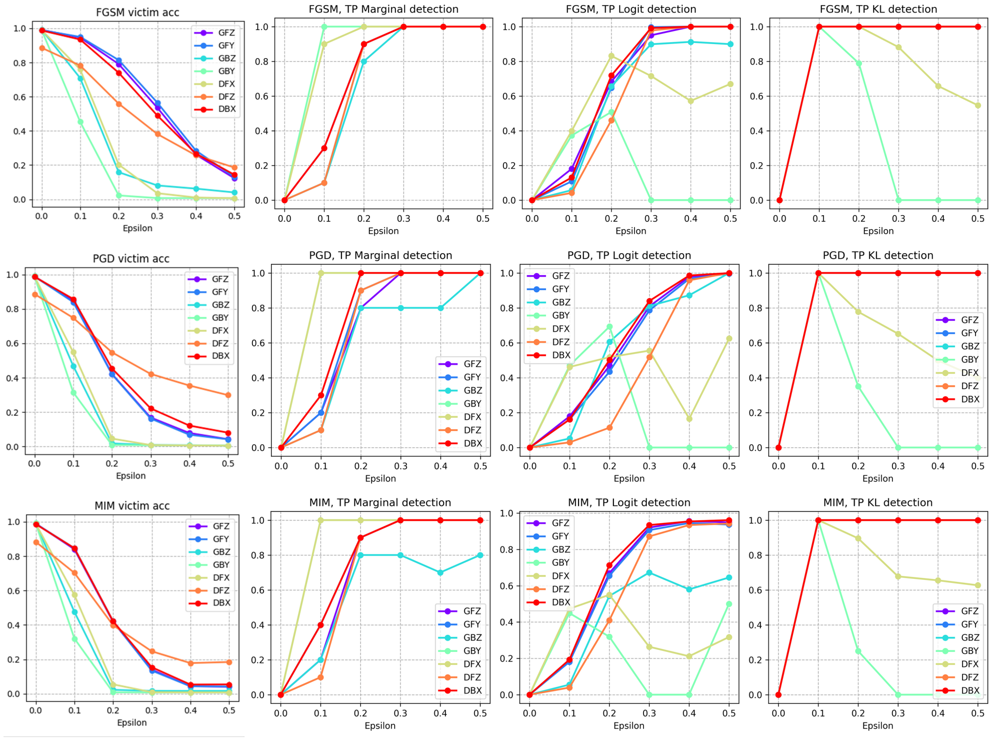

Figure 3. Victim accuracy on the left and detection rate depending on the methods on the right against FGSM, PGD and MIM attack respectively on top, at the middle and at the bottom on MNIST.

uses a constant balancing (c) as a trade-off between imperceptibility and attack success [10].

$$min||x' - x||_2^2 + c \cdot loss(x')$$

5.1.2 Black-box Attacks

We use two simple black-box attacks, called *Gaussian* and *Sticker*. Figure 4 shows an example of such adversarial images.

Gaussian The Gaussian perturbation attack modifies an image by adding Gaussian noise to each pixel. This noise is sampled from a normal distribution with mean $\mu = 0$ and standard deviation $\sigma = \varepsilon$. The perturbation's strength is then controlled by ε.
For an input image $x \in \mathbb{R}^{H \times W \times C}$, the adversarially perturbed image x_{adv} is computed as:

$$x_{\text{adv}} = x + \eta$$

where $\eta \sim \mathcal{N}(\mu, \varepsilon^2)$.
We then rescale x_{adv} to the range $[0, 1]$.

Sticker The sticker attack overlays a visually distinct patch (a sticker) over an image, near its center. The sticker's size ε represents a fraction of the image area. The sticker's color is randomly picked for each test image between different flashy colors: bright yellow, neon green, neon pink, bright cyan, bright orange.

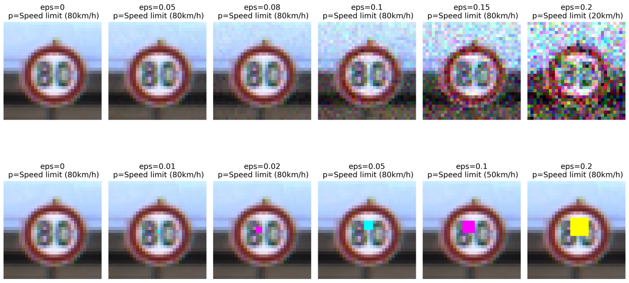

Figure 4. Example adversarial images using black-box attacks Gaussian (top-row) and Sticker (bottom-row), with different ε and the associated prediction from GFZ model.

5.2. Results

We trained the seven models on MNIST following [9], all of them achieving an accuracy $\geq$ 90%. We also trained the seven models on GTSRB, using the same architecture the authors used to train models on CIFAR10. On GTSRB, the models GBZ, GBY and DFX achieved around 95% accuracy, GFZ, GFY and DBX around 50% and around 40% for DFZ. We tested all the models under white-box l_∞ attacks, with $\varepsilon \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ for MNIST. Results are shown in Figure 3.

5.2.1 Results on MNIST

Although we followed precisely and closely the authors procedure using the same hyperparameters to train the models and to implement the adversarial attacks, our results are not similar to [9].

White-box attacks Overall, GFZ and GFY show more robustness for small perturbations than the other models for white-box attacks on MNIST (Figure 3, first column), but are not that robust to more obvious image degradations. Interestingly, a discriminative model, DFZ, shows a good robustness to white-box attacks, but very poor to black-box attacks, especially the Gaussian one.

For detection, on MNIST the results are quite similar to the result of the paper. When $\varepsilon > 0.3$, all the methods are able to detect the attacked images with barely 100% accuracy. However, GBY and DFX do not behave as expected because the detection rate decreases when ε grows, especially with the logit detection and KL-detection. This discrepancy appears to come from the architecture or the model itself, as the marginal method, which relies only on the distribution of the input data and is independent of the classifier's output labels, is not impacted for these two models. In contrast, the logit detection and KL-detection methods, which directly depend on the classifier's predicted labels and probabilities, appear to be more sensitive to adversarial perturbations, likely due to their reliance on the classifier's logits and calibration. However, both models have good accuracy and follow the trend of all their counterparts during the attacks so it is hard to tell why our results are that different from the paper.

Black-box attacks The results for the black-box attacks on MNIST (Figure 6, supplementary material 7) are more in favor of the generative models, that show a less important decrease than their discriminative counterparts with the Gaussian attack's strength. For the sticker attack, it is hard to conclude. This motivates our experiment on GTSRB.

5.2.2 Results on GTSRB

For these experiments, we followed the instructions of the paper. So we chose the same the architecture, same hyperparameters for the attacks and the same training. However, we decided not to take only the well classified images but on the whole dataset, meaning 12,629 images.

First, it is evident that the accuracy of models DFX, GBY, and GBZ, in the absence of attacks, is nearly double that of the other models. However, under adversarial attacks, their accuracy drops significantly more than that of their counterparts, indicating that these models are less robust.

For detection, the results are promising for both marginal and KL-detection methods.Except for the models that underperformed in the previous section, these detection methods can detect 100% of attacked images when $\varepsilon > 0.05$ for KL-detection, or $\varepsilon > 0.1$ for the marginal detection. Nevertheless, the KL-divergence is surprisingly high, even though this is the method with the best results in the paper. This can be attributed to testing the attack on the entire dataset rather than only on correctly classified samples, making the reference probability distribution for the KL-divergence less representative of the data.

In Figure 4, we observe that for $\varepsilon > 0.1$, the adversarial images visibly show the effects of the attack, confirming that these methods accurately reflect reality. In contrast, logit detection performs poorly, achieving only about a 70% detection rate for the highest ε value tested. However, these results are consistent with those reported in the paper.

6. Discussion

In this report, we aimed to reproduce the experiments and findings of [9], investigating whether generative classifiers are indeed more robust to adversarial attacks than their discriminative counterparts. We verified their experiments on MNIST and extended them to a more challenging, real-world dataset: GTSRB. We used white-box and black-box attacks, alongside three detection methods designed to identify adversarial inputs through marginal likelihood estimation, joint likelihood (logits), and divergence-based measures.

On MNIST, we found partial consistency with some of the original paper's conclusions. Certain generative classifiers (GFZ, GFY) did exhibit improved robustness to small adversarial perturbations compared to discriminative models. However, this advantage did not hold for all white-box attack strengths, and the overall robustness trend was less pronounced than reported. Moreover, while the detection methods performed well for large perturbations, the sensitivity of logit and KL-based detection sometimes deviated from expectations, highlighting that the classifiers'

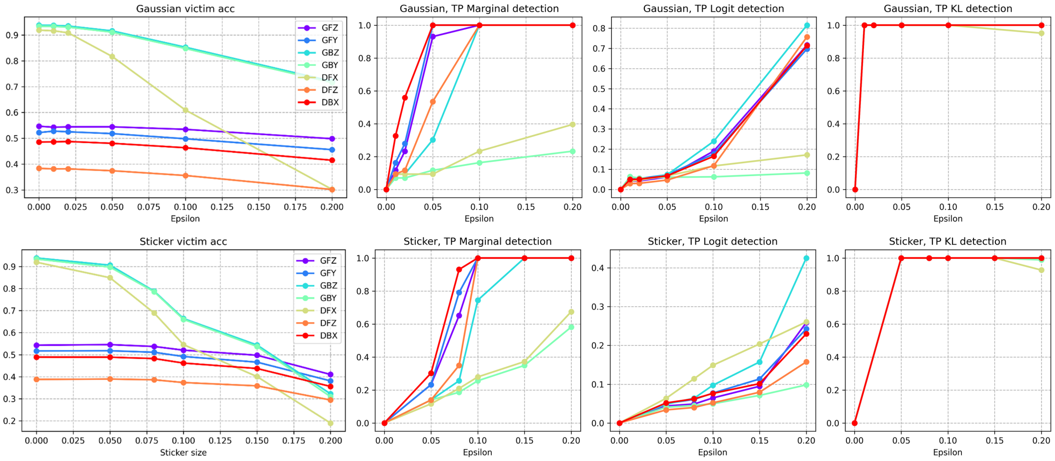

Figure 5. Victim accuracy on the left and detection rate depending on the methods on the right against Gaussian and Sticker attack respectively on top, at the middle and at the bottom on GTSRB.

internal representations may not always generalize well under attack.

When we moved to GTSRB, a dataset with higher complexity and many more classes, the picture became even less clear. Although some generative models were able to maintain slightly higher accuracy against simple black-box perturbations (such as Gaussian noise), we did not observe a definitive, systematic advantage for generative classifiers in more complex scenarios. Moreover, about the detection, both marginal and KL-detection methods show strong performance, detecting 100% of attacked images under certain conditions ($\varepsilon > 0.05$ for KL-detection and $\varepsilon > 0.1$ for marginal detection). However, the unexpectedly high KL-divergence values is not representative of this method's performance. In contrast, logit detection shows poor performance and is not reliable for detecting adversarial images, which aligns with the results of the paper on this detection method.

Our results suggest that the robustness benefits of generative classifiers may be both dataset- and architecture-dependent. While generative approaches can leverage learned data distributions to reject severely off-manifold inputs, their overall resilience to subtle adversarial attacks is not guaranteed. Practical factors, such as the number of classes, the complexity of the input data, and training hyperparameters, may play a significant role in determining whether generative classifiers genuinely confer a robustness advantage over discriminative methods.

In conclusion, we cannot definitively confirm the claims that generative classifiers consistently offer stronger robustness to adversarial attacks. Our experiments underscore the need for further research to disentangle the factors influencing robustness, such as model architecture choices, dataset complexity, and the interplay between detection strategies and classification performance.

References

[1] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks, 2017. 1

[2] Nicholas Carlini and David A. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. *CoRR*, abs/1705.07263, 2017. 3

[3] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: Elastic-net attacks to deep neural networks via adversarial examples, 2018. 1

[4] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 1

[5] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum, 2018. 3

[6] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. 2015. 1, 3

[7] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. 2

[8] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world, 2017. 1

[9] Yingzhen Li, John Bradshaw, and Yash Sharma. Are generative classifiers more robust to adversarial attacks? 2019. 1, 2, 4, 5

[10] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019. 1, 3, 4

[11] Andrew Ng and Michael Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems*. MIT Press, 2001. 1

[12] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroSP)*, pages 372–387, 2016. 1

[13] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, (0):–, 2012. 1

[14] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. 2014. 1

Are Generative Classifiers More Robust to Adversarial Attacks?

Supplementary Material

7. Black-box Results on MNIST

All models behave similarly, except for DFZ which accuracy decreases significantly for the Gaussian attack at the highest ε.

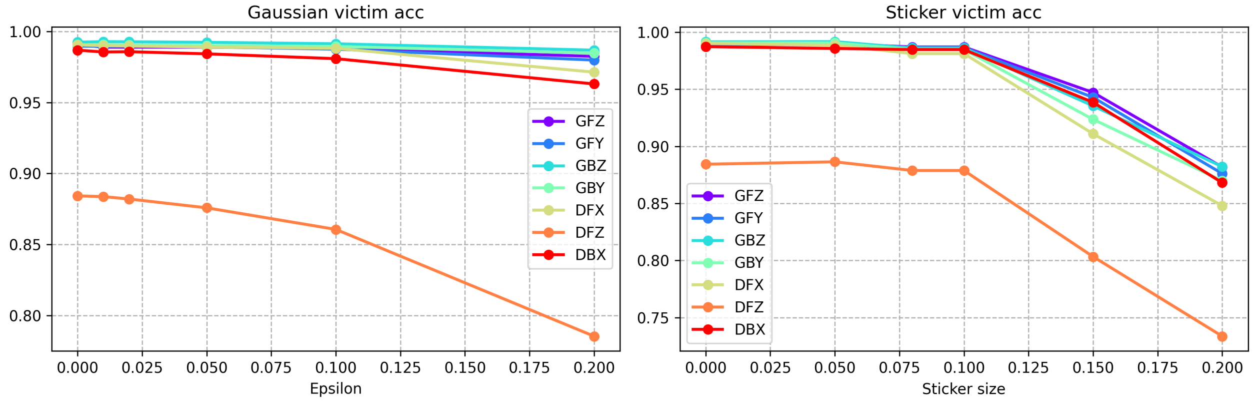

Figure 6. Victim accuracy against black-box Gaussian and Sticker attacks on MNIST. The higher the better