Are Generative Classifiers More Robust to Adversarial Attacks?

Yingzhen Li [1] **John Bradshaw** [2 3] **Yash Sharma** [4]

Abstract

There is a rising interest in studying the robustness of deep neural network classifiers against adversaries, with both advanced attack and defence techniques being actively developed. However, most recent work focuses on *discriminative* classifiers, which only model the conditional distribution of the labels given the inputs. In this paper, we propose and investigate the *deep Bayes* classifier, which improves classical naive Bayes with conditional deep generative models. We further develop detection methods for adversarial examples, which reject inputs with low likelihood under the generative model. Experimental results suggest that deep Bayes classifiers are more robust than deep discriminative classifiers, and that the proposed detection methods are effective against many recently proposed attacks.

1. Introduction

Deep neural networks have been shown to be vulnerable to adversarial examples (Szegedy et al., 2014; Goodfellow et al., 2015). The latest attack techniques can easily fool a deep net with imperceptible perturbations (Goodfellow et al., 2015; Papernot et al., 2016b; Carlini & Wagner, 2017a; Kurakin et al., 2016; Madry et al., 2018; Chen et al., 2018), even in the black-box case, where the attacker does not have access to the network's weights (Papernot et al., 2017b; Chen et al., 2017; Brendel et al., 2018; Athalye et al., 2018; Alzantot et al., 2018a; Uesato et al., 2018). Adversarial attacks are serious security threats to machine learning systems, threatening applications beyond image classification (Carlini & Wagner, 2018; Alzantot et al., 2018b).

To address this outstanding security issue, researchers have proposed defence mechanisms against adversarial attacks. Adversarial training, which augments the training data with adversarially perturbed inputs, has shown moderate success at defending against recently proposed attack techniques (Szegedy et al., 2014; Goodfellow et al., 2015; Tramèr et al., 2018; Madry et al., 2018). In addition, recent advances in Bayesian neural networks have demonstrated that uncertainty estimates can be used to detect adversarial attacks (Li & Gal, 2017; Feinman et al., 2017; Louizos & Welling, 2017; Smith & Gal, 2018). Another notable category of defence techniques involves the usage of generative models. For example, Gu & Rigazio (2014) used an auto-encoder to denoise the inputs before feeding them to the classifier. This denoising approach has been extensively investigated, and the "denoisers" in usage include generative adversarial networks (Samangouei et al., 2018), PixelCNNs (Song et al., 2018) and denoising auto-encoders (Kurakin et al., 2018). These developments rely on the *"off-manifold"* conjecture – adversarial examples are far away from the data manifold, although Gilmer et al. (2018) has challenged this idea with a synthetic "sphere classification" example.

Surprisingly, much less recent work has investigated the robustness of *generative classifiers* (Ng & Jordan, 2002) against adversarial attacks, where such classifiers explicitly model the conditional distribution of the inputs given labels. Typically, a generative classifier produces predictions by comparing between the likelihood of the labels for a given input, which is closely related to the "distance" of the input to the data manifold associated with a class. Therefore, generative classifiers should be robust to many recently proposed adversarial attacks if the "off-manifold" conjecture holds for many real-world applications. Unfortunately, many generative classifiers in popular use, including naive Bayes and linear discriminant analysis (Fisher, 1936), perform poorly on natural image classification tasks, making it difficult to verify the "off-manifold" conjecture and the robustness of generative classifiers with these tools.

Are generative classifiers more robust to recently proposed adversarial attack techniques? To answer this, we improve the naive Bayes algorithm by using deep generative models, and evaluate the conjecture on the proposed generative classifier. In summary, our contributions include:

- We propose *deep Bayes* which models the (conditional) distribution of an input by a deep latent variable model (LVM). We learn the LVM with the variational auto-

[1]Microsoft Research Cambridge, UK [2]University of Cambridge, UK [3]Max Planck Institute for Intelligent Systems, Germany [4]Eberhard Karls University of Tübingen, Germany. Correspondence to: Yingzhen Li <Yingzhen.Li@microsoft.com>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

encoder algorithm (Kingma & Welling, 2014; Rezende et al., 2014), and for classification we approximate Bayes' rule using importance sampling.

- We propose three detection methods for adversarial attacks. The first two use the learned generative model as a proxy of the data manifold, and reject inputs that are far away from it. The third computes statistics for the classifier's output probability vector, and rejects inputs that lead to under-confident predictions.
- We evaluate the robustness of the proposed generative classifier on MNIST and a binary classification dataset derived from CIFAR-10. We further show the advantage of generative classifiers over a number of discriminative classifiers, including Bayesian neural networks and *discriminative* LVMs.
- We improve the robustness of deep neural networks on CIFAR-10 *multi-class* classification, by fusing discriminatively learned visual features with the proposed generative classifiers. On defending a number of popular ℓ_∞ attacks, the fusion model outperforms a baseline discriminative VGG16 network (Simonyan & Zisserman, 2014) by a large margin.

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

Denote $p_{\mathcal{D}}(\boldsymbol{x}, \boldsymbol{y})$ the data distribution for the input $\boldsymbol{x} \in \mathbb{R}^D$ and label $\boldsymbol{y} \in \{\boldsymbol{y}_c | c = 1, ..., C\}$, where $\boldsymbol{y}_c$ is the one-hot encoding vector for class c. For a given $\boldsymbol{x} \in \mathbb{R}^D$ we can define the ground-truth label by $\boldsymbol{y} \sim p_{\mathcal{D}}(\boldsymbol{y}|\boldsymbol{x})$ if $\boldsymbol{x} \in \text{supp}(p_{\mathcal{D}}(\boldsymbol{x}))$. We assume the data distribution $p_{\mathcal{D}}(\boldsymbol{x}, \boldsymbol{y})$ follows the *manifold assumption*: for every class c, the conditional distribution $p_{\mathcal{D}}(\boldsymbol{x}|\boldsymbol{y}_c)$ has its support as a low-dimensional manifold. Thus the training data $\mathcal{D} = \{(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)})\}_{n=1}^N$ is generated as follows: $\boldsymbol{y}^{(n)} \sim p_{\mathcal{D}}(\boldsymbol{y}), \boldsymbol{x}^{(n)} \sim p_{\mathcal{D}}(\boldsymbol{x}|\boldsymbol{y})$.

A generative classifier first builds a *generative model* $p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x}|\boldsymbol{y})p(\boldsymbol{y})$, and then, in prediction time, predicts the label $\boldsymbol{y}^*$ of a test input $\boldsymbol{x}^*$ using Bayes' rule,

$$p(\boldsymbol{y}^*|\boldsymbol{x}^*) = \frac{p(\boldsymbol{x}^*|\boldsymbol{y}^*)p(\boldsymbol{y}^*)}{p(\boldsymbol{x}^*)} = \text{softmax}_{c=1}^C \left[\log p(\boldsymbol{x}^*, \boldsymbol{y}_c)\right],$$

where $\text{softmax}_{c=1}^C$ denotes the softmax operator over the c axis. Here the output probability vector is computed analogously to many discriminative classifiers which use softmax activation in the output layer, so many existing attacks can be tested directly. However, unlike discriminative classifiers, the "logit" values prior to softmax activation have a clear meaning here, which is the log joint distribution $\log p(\boldsymbol{x}^*, \boldsymbol{y}_c)$ of input $\boldsymbol{x}^*$ and a given label $\boldsymbol{y}_c$. Therefore, one can also analyse the logit values to determine whether the unseen pair $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is legitimate, a utility which will be discussed further in section 3.

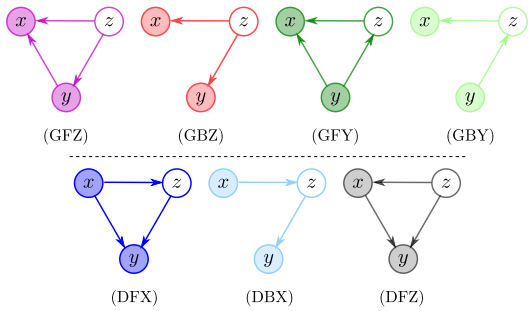


Figure 1. A visualisation of the graphical models, including both **G**enerative (top row) and **D**iscriminative ones (bottom row), as well as **F**ully connected and **B**ottleneck ones. The last character indicates the first node in the topological order of the graph. The graphs are colored in a consistent way as in the result figures.

Naive Bayes is perhaps the most well-known generative classifier; it assumes a factorised distribution for the conditional generator, i.e. $p(\boldsymbol{x}|\boldsymbol{y}) = \prod_{d=1}^D p(x_d|\boldsymbol{y})$, which is inappropriate for e.g. image and speech data. Fortunately, we can leverage the recent advances in generative modelling and apply a deep generative model for the joint distribution $p(\boldsymbol{x}, \boldsymbol{y})$. More specifically, we use a deep latent variable model (LVM) $p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y})$ to construct a generative classifier. Importantly, this leads to a conditional distribution $p(\boldsymbol{x}|\boldsymbol{y}) = \frac{\int p(\boldsymbol{x},\boldsymbol{z},\boldsymbol{y})d\boldsymbol{z}}{\int p(\boldsymbol{x},\boldsymbol{z},\boldsymbol{y})d\boldsymbol{z}d\boldsymbol{x}}$ that is *not* factorised (even when $p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{y})$ is), which is much more powerful than naive Bayes. We refer to such generative classifiers that use deep generative models as *deep Bayes* classifiers.

Depending on the definition of the model $p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y})$, the resulting classifier can be either generative or discriminative. Thus we evaluate the effect of different factorisation structures on the robustness of the induced classifier from the joint distribution $p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y})$ (see Figure 1).

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

We use the initial character "G" to denote generative classifiers and "D" to denote discriminative classifiers. Models with the second character as "F" have a *fully connected* graph, while "B" models have *bottleneck* structures. The last character of the model name indicates the first node in topological order. We do not test other architectures under this nomenclature, as either the graph contains directed cycles (e.g. $\boldsymbol{x} \to \boldsymbol{y} \to \boldsymbol{z} \to \boldsymbol{x}$), or $\boldsymbol{z}$ is the last node in topological order (e.g. $\boldsymbol{x} \to \boldsymbol{y}, (\boldsymbol{x}, \boldsymbol{y}) \to \boldsymbol{z}$) so that the marginalisation of $\boldsymbol{z}$ does not affect classification.

Within the generative classifiers in our design, different graphical models impose different assumptions on the data generation process. E.g. GFZ and GBZ assume there is a confounder $\boldsymbol{z}$ that affects both $\boldsymbol{x}$ and $\boldsymbol{y}$, while GFY and GBY assume the distribution over $\boldsymbol{z}$ is class-dependent. The bottleneck models GBZ, GBY and DBX, when compared with their fully-connected counterparts, enforces the usage of the latent code $\boldsymbol{z}$ for representation learning.

For training, we follow Kingma & Welling (2014) and Rezende et al. (2014) to introduce an amortised approximate posterior $q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y})$, and train both p and q by maximising the *variational lower-bound*:

$$\mathbb{E}_{\mathcal{D}}[\mathcal{L}_{\text{VI}}(\boldsymbol{x}, \boldsymbol{y})] = \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_q \left[\log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n, \boldsymbol{y}_n)}{q(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{y}_n)}\right]. \quad (1)$$

After training, the predicted class probability vector $\boldsymbol{y}^*$ for a future input $\boldsymbol{x}^*$ is computed by an approximation to Bayes' rule with importance sampling $\boldsymbol{z}_c^k \sim q(\boldsymbol{z}|\boldsymbol{x}^*, \boldsymbol{y}_c)$:

$$p(\boldsymbol{y}^*|\boldsymbol{x}^*) \approx \text{softmax}_{c=1}^{C}\left[\log \frac{1}{K}\sum_{k=1}^{K} \frac{p(\boldsymbol{x}^*, \boldsymbol{z}_c^k, \boldsymbol{y}_c)}{q(\boldsymbol{z}_c^k|\boldsymbol{x}^*, \boldsymbol{y}_c)}\right]. \quad (2)$$

Therefore, for generative classifiers the probability of $\boldsymbol{x}$ under the generative model affects the predictions. By contrast, DFX and DBX do not know if $\boldsymbol{x}$ is close to the data manifold or not, as the $p_{\mathcal{D}}(\boldsymbol{x})$ term in these models is cancelled out in eq. (2). Model DFZ is somewhat intermediate, as it builds a generative model for the inputs $\boldsymbol{x}$ (thus $p(\boldsymbol{x}, \boldsymbol{z})$ is used in eq. (2)) but also directly parameterises the conditional distribution $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z})$.

3. Detecting adversarial attacks with generative classifiers

We propose detection methods for adversarial examples using the generative classifier's logit values. As an illustrating example, consider a labelled dataset of "cat" and "dog" images. If an adversarial image of a cat $\boldsymbol{x}_{\text{adv}}$ is incorrectly labelled as "dog", then either this image is ambiguous, or, under a *perfect* generative model, the logit $\log p(\boldsymbol{x}_{\text{adv}}, \text{"dog"})$ will be significantly lower than normal. This means we can detect attacks using the logits $\log p(\boldsymbol{x}^*, \boldsymbol{y}_c), c = 1, ..., C$ computed on a test input $\boldsymbol{x}^*$. The goal here is to reject both unlabelled input $\boldsymbol{x}$ that have low probability under $p(\boldsymbol{x})$ (as a proxy to $p_{\mathcal{D}}(\boldsymbol{x})$), and labelled data $(\boldsymbol{x}, \boldsymbol{y})$ that have low $p(\boldsymbol{x}, \boldsymbol{y})$ values. Concretely, the proposed detection algorithms are as follows.

- **Marginal detection**: rejecting inputs that are far away from the manifold.
  One can select a threshold δ and reject an input $\boldsymbol{x}$ if $-\log p(\boldsymbol{x}) > \delta$. To determine the threshold δ, we can compute $\bar{d}_{\mathcal{D}} = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[-\log p(\boldsymbol{x})]$ and $\sigma_{\mathcal{D}} = \sqrt{\mathbb{V}_{\boldsymbol{x}\sim\mathcal{D}}[\log p(\boldsymbol{x})]}$, then set $\delta = \bar{d}_{\mathcal{D}} + \alpha\sigma_{\mathcal{D}}$.

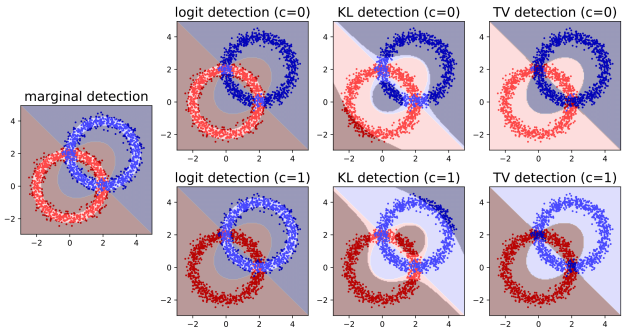


Figure 2. Visualising detection mechanisms. The scattered dots are training data points, with different classes shown in different colours (red for $c = 0$ and blue for $c = 1$). Same labels are manually assigned for inputs when the detection method requires $\boldsymbol{y}$. Decision regions are shown in the corresponding colours. Input points in the shaded area are rejected by detection methods.

- **Logit detection**: rejecting inputs using joint density.
  Given a victim model $\boldsymbol{y} = F(\boldsymbol{x})$, one can reject $\boldsymbol{x}$ if $-\log p(\boldsymbol{x}, F(\boldsymbol{x})) > \delta_{\boldsymbol{y}}$. We can use the mean and variance statistics $\bar{d}_c, \sigma_c$ computed on $\log p(\boldsymbol{x}, \boldsymbol{y}_c)$ and select $\delta_{\boldsymbol{y}_c} = \bar{d}_c + \alpha\sigma_c$.
- **Divergence detection**: rejecting inputs with over- and/or under-confident predictions.
  Denote $\mathbf{p}(\boldsymbol{x})$ as a C-dimensional probability vector outputted by the classifier. For each class c, we first collect the *mean classification probability vector* $\mathbf{p}_c = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}_c)\in\mathcal{D}}[\mathbf{p}(\boldsymbol{x})]$, then compute the mean $\bar{d}_c$ and standard deviation σ_c on a selected divergence/distance measure $\mathrm{D}[\mathbf{p}_c||\mathbf{p}(\boldsymbol{x})]$ for all $(\boldsymbol{x}, \boldsymbol{y}_c) \in \mathcal{D}$. A test input $\boldsymbol{x}^*$ with prediction label $c^* = \arg\max \mathbf{p}(\boldsymbol{x}^*)$ is rejected if $\mathrm{D}[\mathbf{p}_{c^*}||\mathbf{p}(\boldsymbol{x}^*)] > \bar{d}_{c^*} + \alpha\sigma_{c^*}$. Therefore, an example $\boldsymbol{x}^*$ can be rejected if the probability vector $\mathbf{p}(\boldsymbol{x}^*)$ is very different to the ones seen in training.

  When D is the KL-divergence, we call this method *KL detection*. Other divergence/distance measures such as total variation (TV) can also be used.

For better intuition, we visualise the detection mechanisms in Figure 2 with a synthetic "two rings" binary classification example. In this case we sample 2×1000 training data points by $\boldsymbol{y} \sim \text{Bern}(0.5), \theta \sim \text{Uniform}(0, 2\pi), \boldsymbol{x}|\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{x}; \boldsymbol{c}_{\boldsymbol{y}} + r_{\boldsymbol{y}}[cos(\theta), sin(\theta)]^{\mathrm{T}}, \sigma^2\mathbf{I})$. We consider a generative classifier $p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x}|\boldsymbol{y})p_{\mathcal{D}}(\boldsymbol{y}), p(\boldsymbol{x}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_{\boldsymbol{y}}, \sigma^2\mathbf{I}), \boldsymbol{\mu}_{\boldsymbol{y}} = \arg\min_{||\hat{\boldsymbol{x}}-\boldsymbol{c}_{\boldsymbol{y}}||_2=r_{\boldsymbol{y}}} ||\boldsymbol{x} - \hat{\boldsymbol{x}}||_2$. The δ thresholds are selected to achieve 10% false positive rates on training data. From the visualisations we see that inputs that are far away from the model manifold are rejected by marginal/logit detection. At the same time, logit detection rejects data points that are not on the manifold of the given class. KL/TV detection does not construct manifold-aware acceptance regions, which is as expected since the proposed divergence detection method does not require the classifier to be generative. However, both detection methods have

some success in rejecting uncertain predictions, especially for TV, which also rejects ambiguous inputs (see the ring-cross regions in the last two plots). Combining all three methods, we see that the rejected inputs are either far away from the manifold, or are ambiguous.

Detection methods using logit values have been used in e.g. Li & Gal (2017); Feinman et al. (2017), but it is unclear whether the logits values in discriminative classifiers have a semantic meaning. Song et al. (2018); Samangouei et al. (2018); Kurakin et al. (2018) trained a *separate* generative model for denoising/detection. But the features of the generator and the discriminative classifier can be very different, hence the generator cannot detect the "manifold attack" against the classifier (Gilmer et al., 2018). Unlike these approaches, we highlight three critical properties of generative classifiers and the accompanied detection methods:

1. the representations of the data manifold are the same for both the classifier and the detector (since they share the same generative model $p(\boldsymbol{x}, \boldsymbol{y}) \approx p_{\mathcal{D}}(\boldsymbol{x}, \boldsymbol{y})$;
2. the logit values in generative classifiers have a clear semantic meaning: the log probability $\log p(\boldsymbol{x}, \boldsymbol{y})$ of generating the input $\boldsymbol{x}$ given the class label $\boldsymbol{y} = \boldsymbol{y}_c$; also note that $F(\boldsymbol{x}) = \arg\max_{\boldsymbol{y}_c} \log p(\boldsymbol{x}, \boldsymbol{y}_c)$;
3. the marginal/logit detection aims at rejecting $\boldsymbol{x}$ that has low $\log p(\boldsymbol{x})$ and/or $\log p(\boldsymbol{x}, \boldsymbol{y})$. These probabilities are obtained as part of the classification procedure and so do not require the running of any extra models.

Assuming an accurate approximation $p(\boldsymbol{x}, \boldsymbol{y}) \approx p_{\mathcal{D}}(\boldsymbol{x}, \boldsymbol{y})$, a new input $\boldsymbol{x}^*$ is accepted by marginal/logit detection only if $\boldsymbol{x}^*$ is close to the manifold of $F(\boldsymbol{x}^*)$ (thus $p(\boldsymbol{x}^*, \boldsymbol{y} = F(\boldsymbol{x}^*))$ is high).[1] So $F(\boldsymbol{x}^*)$ should be equal to the ground truth label $\boldsymbol{y}^*$ if $\boldsymbol{x}^*$ is not an ambiguous input (which will be detected by KL/TV detection). Therefore the "off-manifold" conjecture should hold for a powerful generative classifier, and below we present an empirical study to validate the assumptions of this conjecture.

4. Experiments

We carry out a number of tests on the deep Bayes classifiers, our implementation is available at `https://github.com/deepgenerativeclassifier/DeepBayes`.

The distributions $q(\boldsymbol{z}|\cdot)$ and $p(\boldsymbol{z}|\cdot)$ are factorised Gaussians, and the conditional probability $p(\boldsymbol{x}|\cdot)$, if required, is parameterised by an ℓ_2 loss. Besides the LVM-based classifiers, we further train a Bayesian neural network (BNN) with Bernoulli dropout (dropout rate 0.3), as it has been shown in Li & Gal (2017) and Feinman et al. (2017) that BNNs are more robust than their deterministic counterparts. The constructed BNN has 2x more channels than LVM-based classifiers, making the comparison slightly "unfair", as the BNN layers have more capacity. We use $K = 10$ Monte Carlo samples for all the classifiers.

The adversarial attacks are taken from the CleverHans 2.0 library (Papernot et al., 2017a). Three metrics are reported: *accuracy* of the classifier on crafted adversarial examples, mean *minimum perturbation distance* computed on adversarial examples that have successfully fooled the classifier, and *detection rate* on successful attacks. This detection rate is defined as the true positive (TP) rate of finding an adversarial example, and the detection threshold is selected to achieve a 5% false positive rate on clean training data.

The experiments are performed under various threat model settings. In the main text we present gradient-based attacks and gradient masking sanity checks, and provide a brief summary of further experiments presented in the appendix. Full table results can also be found in appendix F.

4.1. Gradient-based attacks

We first evaluate the robustness of generative classifiers against gradient-based attacks. These attacks are performed under a white-box setting *against the classifier* (Carlini & Wagner, 2017b): the attacker can differentiate through the classifier, but is not aware of the existence of the detector. We report results on two datasets (MNIST and CIFAR-binary) and two classes of attacks (ℓ_∞ and ℓ_2 attacks).

Datasets For MNIST tests, we use $\dim(\boldsymbol{z}) = 64$ for the LVM-based classifiers. All classifiers achieve $> 98\%$ clean test accuracy, therefore we apply attacks on the whole test dataset (10,000 datapoints). Since the robustness properties of MNIST classifiers might not extend to natural images (Carlini & Wagner, 2017b), we further consider the same set of evaluations on *CIFAR-binary*, a binary classification dataset containing "airplane" and "frog" images from CIFAR-10. We choose to work with this simpler dataset because (1) *fully* generative classifiers are less satisfactory for classifying clean CIFAR-10 images[2], and (2) we want to evaluate whether the robustness properties of *fully* generative classifiers hold on natural images (c.f. Carlini & Wagner, 2017b). On CIFAR-binary, the generative classifiers use $\dim(\boldsymbol{z}) = 128$ and obtain $> 90\%$ clean test accuracy (see appendix). The attacks are then performed on the test images that all models initially correctly classify, leading to a test set of 1,577 instances for CIFAR-binary. For both datasets the images are scaled to [0, 1].

[1] Nalisnick et al. (2019) showed that existing generative models fail to detect all possible outliers. A solution to counter this issue in logit/marginal detection is to add a lower-bounding threshold so that an input-output pair $(\boldsymbol{x}, \boldsymbol{y})$ will be rejected if $-\log p(\boldsymbol{x}, \boldsymbol{y}) < \zeta_{\boldsymbol{y}}$. We leave this investigation to future work.

[2] The clean test accuracies for GFZ & GFY on CIFAR-10 are all $< 50\%$; a conditional PixelCNN++ (Salimans et al., 2017) (with much deeper networks) achieves 72.4% clean test accuracy.

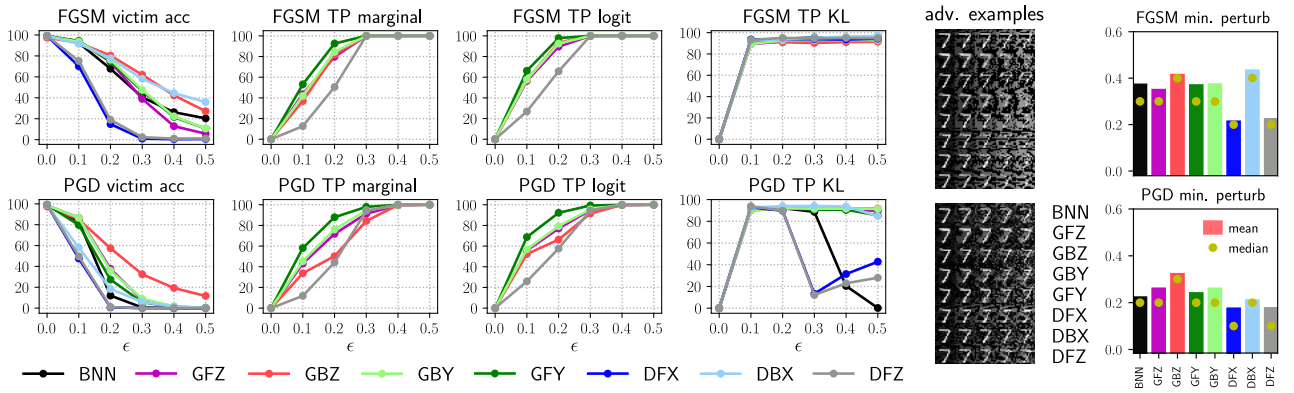


Figure 3. Victim accuracy, detection rates and minimum ℓ_{inf} perturbation against **white-box FGSM attacks** on MNIST. The higher the better. The visualised adversarial examples (not necessarily successful) are crafted with with ℓ_∞ distortion ϵ growing from 0.1 to 0.5.

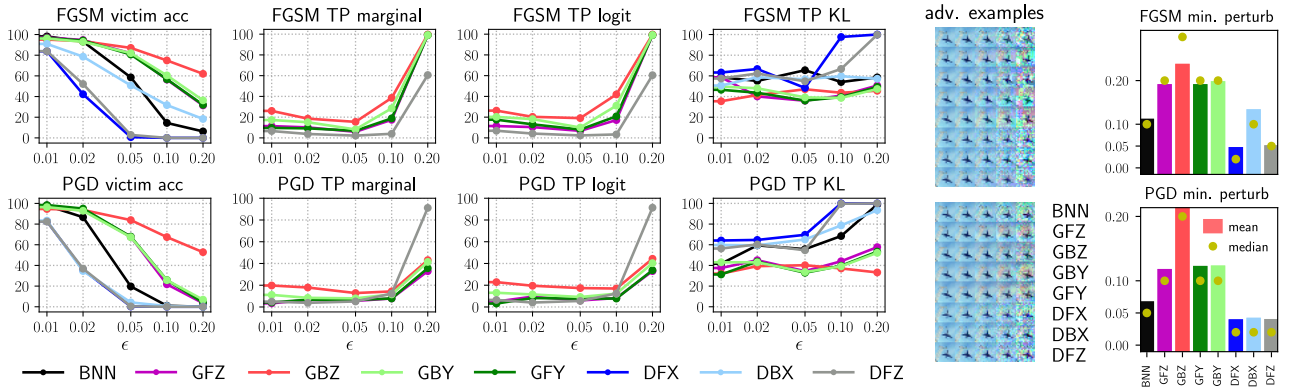


Figure 4. Victim accuracy, detection rates and minimum ℓ_{inf} perturbation against **white-box FGSM attacks** on CIFAR-binary. The higher the better. The visualised adversarial examples (not necessarily successful) are crafted with ℓ_∞ distortion ϵ growing from 0.01 to 0.2.

ℓ_∞ attacks The attacks in test are: fast gradient sign method (FGSM, Goodfellow et al., 2015), projected gradient descent (PGD, Madry et al., 2018) and momentum iterative method (MIM, Dong et al., 2018).[3] We use the distortion strengths as $\epsilon \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ for MNIST, and $\epsilon \in \{0.01, 0.02, 0.05, 0.1, 0.2\}$ for CIFAR-binary.

Results are reported in Figure 3 for MNIST and Figure 4 for CIFAR-binary, respectively. For both datasets, generative classifiers perform generally better in terms of victim accuracy, especially GBZ is significantly more robust than the others on CIFAR-binary. By contrast, discriminative VAE-based classifiers are less robust, e.g. on MNIST, DFX & DFZ are not robust to the weakest attack (FGSM) even when $\epsilon = 0.2$ (where the distorted inputs are still visually close to the original digit "7"). Interestingly, DBX is relatively robust against FGSM on both datasets, which agrees with the preliminary tests in Alemi et al. (2017). Further investigations in appendix A.2 show that the bottleneck structure might be beneficial for defending certain attacks.

In terms of minimum perturbation which is computed across all ϵ settings,[4] quantitatively the amount of distortion required to fool generative classifiers is much higher than that for discriminative ones. Indeed on both datasets, the visual distortion of the adversarial examples on generative classifiers is more significant.

For detection, generative classifiers successfully detect the adversarial examples with $\epsilon \geq 0.3$ on MNIST, which is reasonable as the visual distortion is already significant. Detection results on CIFAR-binary are less satisfactory though: marginal/logit detection fail to detect attacks with $\epsilon = 0.1$ (which attain both high success rate and induce visually perceptible distortion). These results suggest that the per-pixel ℓ_2 loss might not be best suited for modelling natural images (c.f. Larsen et al., 2016; van den Oord et al., 2016), indeed we present improved robustness results in section 4.3, where the generative classifiers use an alternative likelihood function that is closely related to the *perceptual loss* (Dosovitskiy & Brox, 2016; Johnson et al., 2016).

[3] See appendix B for MIM results with similar observations as in the main text.

[4] We manually assign the minimum perturbation of an input as $\epsilon_{\text{max}} + 0.1$ if none of the attacks is successful.

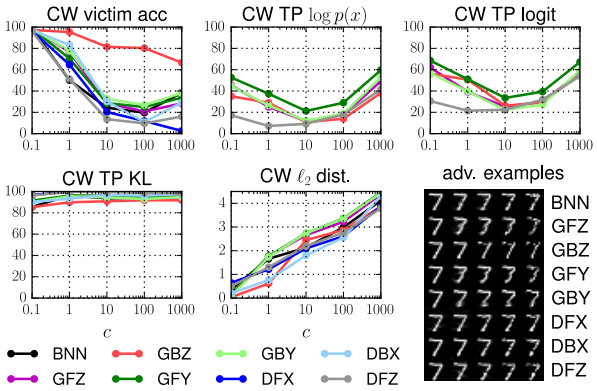
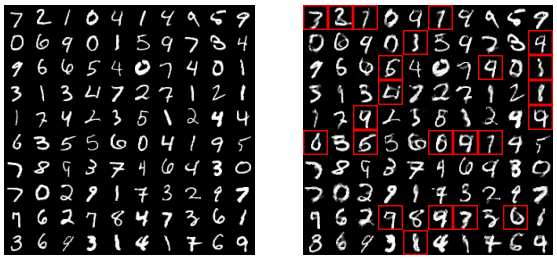


Figure 5. Accuracy, ℓ_2 distortion, and detection rates against **white-box CW attacks** on MNIST.



(a) clean inputs (b) CW adv. inputs

Figure 6. Clean inputs and CW adversarial examples ($c = 10$) crafted on GFZ, digits in red rectangles show significant ambiguity.

As a side note, DFZ, as an intermediate between generative and discriminative classifiers, has worse robustness results, but has good detection performance for the marginal and logit metrics. This is because with softmax activation, the marginal distribution $p(\boldsymbol{x})$ is dropped, but in marginal/logit detection $p(\boldsymbol{x})$ is still in use. KL detection works well for all classifiers, and on CIFAR-binary, discriminative classifiers start to dominate in this metric as ϵ increases. Inspecting the visualised adversarial examples on generative classifiers (Figure 4), we see that for large ϵ values, these inputs are significantly distorted, thus "not ambiguous", indeed they are detected by marginal/logit detection methods.

ℓ_2 attack We perform the Carlini & Wagner (CW) ℓ_2 attack (Carlini & Wagner, 2017a), with a loss-balancing parameter $c \in \{0.1, 1, 10, 100, 1000\}$, so that the ℓ_2 distortion regulariser in CW's loss function decreases with larger c.

MNIST results are reported in Figure 5. Here GBZ is a clear winner: for a given level of ℓ_2 distortion, GBZ is significantly more robust than the others. Other classifiers perform similarly on MNIST, however, these attack successes on generative classifiers is mainly due to the ambiguity of the crafted adversarial images. As visualised in Figure 6 (and Figure B.4 in appendix), the induced distortion from CW leads to ambiguous digits which sit at the perceptual boundary between the original and the adversarial classes.

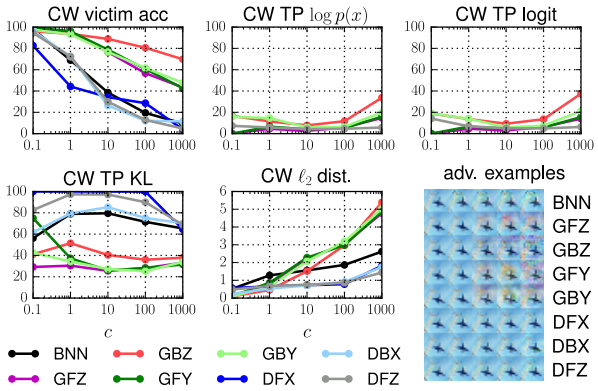


Figure 7. Accuracy, ℓ_2 distortion, and detection rates against **white-box CW attacks** on CIFAR-binary.

Interestingly, the detection rates on MNIST adversarial inputs do not grow monotonically as c increases. Combined with the victim accuracy results, this means $c = 10$ is the sweet-spot parameter that achieves the best success rates against both the classifier and the marginal/logit detection methods. KL detection achieves $> 95\%$ detection rates on all c and all the classifiers. This is as expected as the CW attack generates adversarial examples that lead to minimal difference between the logit values of the most and the second most probable classes. In this case the adversarial examples might correspond to ambiguous inputs (Figure 6).

Figure 7 presents the robustness and detection results for CIFAR-binary. Here the generative classifiers are significantly more robust than the others (with the best being GBZ), and the mean ℓ_2 distortions computed on successful attacks are also significantly higher. The TP rates are low for marginal/logit detection when c is small, which is reasonable as the crafted images are visually similar to the clean ones. Note that the distortion for the attacks on generative classifiers is perceptible, and the logit TP rates also increase as c increases. These results indicate that this CW attack is ineffective when attacking generative classifiers.

Summary The tested generative classifiers are generally more robust than the discriminative ones against white-box gradient-based attacks. In particular, the generative classifier's victim accuracy decreases as the distortion increases, but at the same time the TP rates for marginal/logit detection also increase. Therefore the two attacks in test fail to find near-manifold adversarial examples that fool both the classifier and the detector components of generative classifiers.

4.2. Sanity checks on gradient masking

If a successful defence against white-box gradient-based attacks is due to gradient masking, then this defence is likely to be less effective against attacks that do not differentiate through the victim classifier and the defence (Papernot et al.,

Table 1. Mean minimum ℓ_∞ perturbation (in red, computed on $\epsilon \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$) and victim accuracy (in blue, for $\epsilon \leq 0.3$) for ℓ_∞ attacks on MNIST.

Attack	GFZ	GBZ	GFY	GBY
PGD (white)	0.23 / 7.71%	0.30 / 30.78%	0.21 / 5.52%	0.23 / 8.89%
MIM (white)	0.24 / 9.02%	0.21 / 4.97%	0.22 / 6.72%	0.21 / 1.54%
PGD (grey)	0.37 / 51.08%	0.36 / 50.64%	0.38 / 53.29%	0.36 / 48.66%
MIM (grey)	0.34 / 43.00%	0.33 / 40.94%	0.34 / 46.64%	0.33 / 40.06%
PGD (black)	0.40 / 61.93%	0.42 / 66.75%	0.38 / 56.35%	0.43 / 68.50%
MIM (black)	0.36 / 50.44%	0.38 / 59.86%	0.36 / 48.07%	0.39 / 61.78%

Table 2. Mean minimum ℓ_∞ perturbation (in red, computed on $\epsilon \in \{0.01, 0.02, 0.05, 0.1, 0.2\}$) and victim accuracy (in blue, for $\epsilon \leq 0.1$) for ℓ_∞ attacks on CIFAR-binary.

Attack	GFZ	GBZ	GFY	GBY
PGD (white)	0.11 / 21.81%	0.20 / 65.63%	0.11 / 25.81%	0.11 / 25.24%
MIM (white)	0.09 / 15.22%	0.13 / 37.60%	0.10 / 16.4%9	0.09 / 14.39%
PGD (grey)	0.15 / 50.48%	0.23 / 77.30%	0.16 / 54.66%	0.17 / 57.96%
MIM (grey)	0.15 / 47.62%	0.21 / 75.71%	0.15 / 51.11%	0.16 / 53.84%
PGD (black)	0.19 / 68.36%	0.23 / 79.45%	0.20 / 70.13%	0.19 / 67.98%
MIM (black)	0.18 / 66.39%	0.23 / 78.38%	0.19 / 68.42%	0.19 / 66.52%

2017b; Athalye et al., 2018). Therefore, here we present two types of attacks as sanity checks on gradient masking.

Distillation-based attacks We perform two attacks based on distilling the victim classifier using a "student" CNN. The two attacks differ in their threat models: in the **grey-box** setting the attacker has access to both the training data and the output probability vectors of the classifiers on the training set, while in the **black-box** setting the attacker only has access to queried labels on a given input. For the latter black-box setting, we follow Papernot et al. (2017b) to train a substitute CNN using Jacobian-based dataset augmentation (see appendix E.1). We then craft adversarial examples on the grey-/black-box substitutes using PGD and MIN, and transfer them to the victim classifiers. As a sanity check, these attacks with reasonably small ϵ values achieve $\sim 100\%$ success rates on fooling the substitutes ($\epsilon \geq 0.2$ for MNIST and $\epsilon \geq 0.1$ for CIFAR-binary, see appendix F).

We report the results for these transferred attacks in Table 1 for MNIST and Table 2 for CIFAR-binary, respectively, with a comparison to white-box attack results taken from the last section. It is clear that the adversarial examples crafted on substitute models do not transfer very well to the generative classifiers. Importantly, for a fixed ϵ setting, the white-box attacks achieve significantly higher success rates than their grey-/black-box counterparts, and the gap is at least $> 20\%$ for MNIST (with $\epsilon \leq 0.3$) and $> 30\%$ for CIFAR-binary (with $\epsilon \leq 0.1$). Furthermore, the mean minimum ℓ_∞ perturbation obtained by grey-/black-box attacks is significantly higher than those obtained by white-box attacks.

SPSA (evolutionary strategies) We consider another black-box setting that only assumes access to the logit values of the prediction given an input. We use the SPSA ℓ_∞ attack (Uesato et al., 2018), which is similar to the white-box attacks, except that gradients are numerically estimated using the logit values from the victim classifier. Results are presented in Table 3 for MNIST (using 1,000 randomly sampled test datapoints) and CIFAR-binary. Both experiments clearly show that SPSA performs much worse on generative classifiers when compared to *white-box* PGD.

Table 3. Victim accuracy results against **white-box PGD** and **black-box SPSA attack**. We use $\epsilon = 0.3$ for MNIST and $\epsilon = 0.05$ for CIFAR-binary.

	MNIST		CIFAR-binary	
	PGD	SPSA	PGD	SPSA
GFZ	4.0%	68.2%	67.7%	96.4%
GBZ	29.7%	79.0%	83.9%	95.2%
GBY	7.4%	71.0%	67.9%	96.4%
GFY	2.3%	55.9%	67.3%	96.3%

Summary Both distillation-based attacks and score-based attack (SPSA) failed to obtain higher success rate than gradient-based attacks against generative classifiers. Therefore, gradient masking is unlikely to be responsible for the improved robustness of generative classifiers, as utilising the exact gradient yielded better success rates for the attacks.

4.3. Combining deep Bayes and discriminative features

This experiment examines the robustness of CIFAR-10 *multi-class* classifiers, with the generative classifiers trained on *discriminative* visual features. To do this, we download a pretrained VGG16 network (Simonyan & Zisserman, 2014) on CIFAR-10 (93.59% test accuracy),[5] and use its features $\boldsymbol{\phi}(\boldsymbol{x})$ as the input to the VAE-based classifiers: $p(\boldsymbol{y}|\boldsymbol{x}) = p(\boldsymbol{y}|\boldsymbol{\phi}(\boldsymbol{x}))$. This means $p(\boldsymbol{x}|\cdot)$ of the generative classifiers is defined by a perceptual loss (Dosovitskiy & Brox, 2016; Johnson et al., 2016), see appendix C for a discussion. The classifiers in test include GBZ, GBY and DBX. We use fully-connected neural networks for these classifiers, and select from VGG16 the 9^{th} convolution layer (CONV9) and the first fully connected layer after convolution (FC1) as the feature layers to ensure $\sim 90\%$ test accuracy.

Results on white-box ℓ_∞ attacks are visualised in Figure 8. For all LVM-based classifiers we see clear improvements in robustness and detection over the VGG16 baseline. In particular, GBZ and GBY with CONV9 features are overall better than DBX. More importantly, generative classifiers based on CONV9 features are significantly more robust than those based on FC1 features. By contrast, for DBX, which is *discriminative*, the robustness results are very similar. This indicates that the level of feature representation has little effect for DBX, presumably DBX-CONV9 has learned high-level features that resembles FC1. Also the logit detection method

[5] https://github.com/geifmany/cifar-vgg

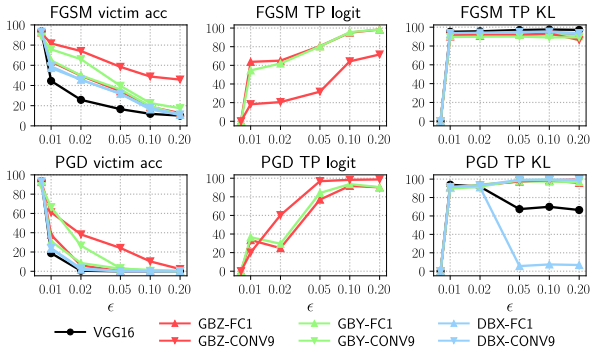


Figure 8. Accuracy and detection rates against **white-box** ℓ_∞ attacks on CIFAR-10. The higher the better. Note that results for DBX, GBZ-FC1 and GBY-FC1 are almost identical.

works much better on the fusion models when compared with the generative classifiers using ℓ_2 likelihood (c.f. Figure 4). These results suggest that one can achieve both high clean accuracy and better robustness/detection rates against adversaries by combining discriminatively learned visual features and generative classifiers.

4.4. Summary of additional studies

We present in appendix A further studies on the robustness properties of generative and discriminative classifiers.

- In appendix A.1, we designed a white-box attack targeting both the classifier and the detector, which also considers the usage of random $\boldsymbol{z}$ samples by the LVM-based classifiers (Biggio et al., 2013; Carlini & Wagner, 2017b). In results, although this attack can reduce detection levels, it comes with the trade-off of increasing accuracy, suggesting that this adversary cannot break both the classifier and detector working in tandem.
- In appendix A.2, we quantified the bottleneck effect by varying dim($\boldsymbol{z}$) in **B**ottleneck classifiers. Results indicate that using a small bottleneck improves the robustness of the classifier against ℓ_∞ attacks.
- In appendix A.3, we evaluated the transferability of adversarial examples accross different LVM-based classifiers. We found that these transferred attacks are relatively effective between generative classifiers, but not from generative to discriminative (and vice versa).

5. Discussion

We have proposed *deep Bayes* as a generative classifier that uses deep LVMs to model the joint distribution of input-output pairs. We have given evidence that generative classifiers are more robust to many recent adversarial attacks than discriminative classifiers. Furthermore, the logit in generative classifiers has a well-defined meaning and can be used to detect attacks, even when the classifier is fooled.

Concurrent to us, Schott et al. (2019) also demonstrated the robustness of generative classifiers on MNIST, in which their graphical model corresponds to GFY in our design, and the logits are computed by a tempered version of the variational lower-bound. However, their approach requires thousands of random $\boldsymbol{z}$ samples and tens of optimisation steps to approximate $\log p(\boldsymbol{x}|\boldsymbol{y})$ for every input-output pair $(\boldsymbol{x}, \boldsymbol{y})$, making it much less scalable than our importance sampling technique. Indeed, we have scaled our approach to CIFAR-10, a natural image dataset, and the robustness results are consistent with those on MNIST.

Importantly, the graphical model structure has a significant impact on robustness, which is not mentioned in Schott et al. (2019). Our study shows that deep LVM-based generative classifiers generally outperform the (randomised) discriminative ones, and the bottleneck is useful for defending against ℓ_∞ attacks. Our results corroborate with the Bayesian neural network literature, in particular Li & Gal (2017); Feinman et al. (2017); Carlini & Wagner (2017b), in showing that modelling unobserved variables are effective for defending against adversarial attacks.

While we have given strong evidence to suggest that generative classifiers are more robust to current adversarial attacks, we do not wish to claim that these models will be robust to *all* possible attacks. Aside from many recent attacks being designed specifically for discriminative neural networks, there is also evidence for the fragility of generative models; e.g. naive Bayes as a standard approach for spam filtering is well-known to be fragile (Dalvi et al., 2004; Huang et al., 2011), and recently Tabacof et al. (2016); Kos et al. (2017); Creswell et al. (2017) also designed attacks for (unconditional) VAEs. However, generative classifiers can be made more robust too, to counter these weaknesses. Dalvi et al. (2004) have shown that generative classifiers can be made more secure if aware of the attack strategy, and Biggio et al. (2011; 2014) further improved naive Bayes' robustness by modelling the conditional distribution of the adversarial inputs. These approaches are similar to the adversarial training of discriminative classifiers (Tramèr et al., 2018; Madry et al., 2018), efficient ways for doing so with generative classifiers can be an interesting research direction.

But even with this note of caution, we believe this work offers exciting avenues for future work. Using generative classifiers offers an interesting way to evaluate generative models and can drive improvements in their ability to tackle high-dimensional datasets, where traditionally generative classifiers have been less accurate than discriminative classifiers (Efron, 1975; Ng & Jordan, 2002). In addition, the combination of generative and discriminative models is a compelling direction for future research. Overall, we believe that progress on generative classifiers can inspire better designs of attack, defence and detection techniques.

Acknowledgements

John Bradshaw acknowledges support from an EPSRC studentship.

References

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017.

Alzantot, M., Sharma, Y., Chakraborty, S., and Srivastava, M. Genattack: Practical black-box attacks with gradient-free optimization. *arXiv preprint arXiv:1805.11090*, 2018a.

Alzantot, M., Sharma, Y., Elgohary, A., Ho, B.-J., Srivastava, M., and Chang, K.-W. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998*, 2018b.

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, pp. 274–283, 2018.

Biggio, B., Fumera, G., and Roli, F. Design of robust classifiers for adversarial environments. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pp. 977–982. IEEE, 2011.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 387–402. Springer, 2013.

Biggio, B., Fumera, G., and Roli, F. Security evaluation of pattern classifiers under attack. *IEEE transactions on knowledge and data engineering*, 26(4):984–996, 2014.

Brendel, W., Rauber, J., and Bethge, M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SyZI0GWCZ.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pp. 39–57. IEEE, 2017a.

Carlini, N. and Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 3–14. ACM, 2017b.

Carlini, N. and Wagner, D. Audio adversarial examples: Targeted attacks on speech-to-text. *arXiv preprint arXiv:1801.01944*, 2018.

Chen, P. Y., Zhang, H., Sharma, Y., Yi, J., and Hsieh, C. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 15–26. ACM, 2017.

Chen, P.-Y., Sharma, Y., Zhang, H., Yi, J., and Hsieh, C.-J. Ead: Elastic-net attacks to deep neural networks via adversarial examples, 2018.

Creswell, A., Bharath, A. A., and Sengupta, B. Latentpoison-adversarial attacks on the latent space. *arXiv preprint arXiv:1711.02879*, 2017.

Dalvi, N., Domingos, P., Sanghai, S., Verma, D., et al. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 99–108. ACM, 2004.

Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., and Li, J. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9185–9193, 2018.

Dosovitskiy, A. and Brox, T. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems*, pp. 658–666, 2016.

Efron, B. The efficiency of logistic regression compared to normal discriminant analysis. *Journal of the American Statistical Association*, 70(352):892–898, 1975.

Feinman, R., Curtin, R. R., Shintre, S., and Gardner, A. B. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

Fisher, R. A. The use of multiple measurements in taxonomic problems. *Annals of human genetics*, 7(2):179–188, 1936.

Gatys, L. A., Ecker, A. S., and Bethge, M. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2414–2423, 2016.

Gilmer, J., Metz, L., Faghri, F., Schoenholz, S. S., Raghu, M., Wattenberg, M., and Goodfellow, I. Adversarial spheres. *arXiv preprint arXiv:1801.02774*, 2018.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

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

Gu, S. and Rigazio, L. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014.

Huang, L., Joseph, A. D., Nelson, B., Rubinstein, B. I., and Tygar, J. Adversarial machine learning. In *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, pp. 43–58. ACM, 2011.

Johnson, J., Alahi, A., and Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pp. 694–711. Springer, 2016.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hk99zCeAb.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Kos, J., Fischer, I., and Song, D. Adversarial examples for generative models. *arXiv preprint arXiv:1702.06832*, 2017.

Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

Kurakin, A., Goodfellow, I., Bengio, S., Dong, Y., Liao, F., Liang, M., Pang, T., Zhu, J., Hu, X., Xie, C., et al. Adversarial attacks and defences competition. *arXiv preprint arXiv:1804.00097*, 2018.

Larsen, A. B. L., Sønderby, S. K., Larochelle, H., and Winther, O. Autoencoding beyond pixels using a learned similarity metric. In *International Conference on Machine Learning*, pp. 1558–1566, 2016.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al. Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 105–114. IEEE, 2017.

Li, Y. and Gal, Y. Dropout inference in Bayesian neural networks with alpha-divergences. In *International Conference on Machine Learning*, pp. 2052–2061, 2017.

Louizos, C. and Welling, M. Multiplicative normalizing flows for variational Bayesian neural networks. In *International Conference on Machine Learning*, pp. 2218–2227, 2017.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1xwNhCcYm.

Ng, A. Y. and Jordan, M. I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems*, pp. 841–848, 2002.

Papernot, N., McDaniel, P., and Goodfellow, I. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016a.

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pp. 372–387. IEEE, 2016b.

Papernot, N., Carlini, N., Goodfellow, I., Feinman, R., Faghri, F., Matyasko, A., Hambardzumyan, K., Juang, Y.-L., Kurakin, A., Sheatsley, R., Garg, A., and Lin, Y.-C. cleverhans v2.0.0: an adversarial machine learning library. *arXiv preprint arXiv:1610.00768*, 2017a.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pp. 506–519. ACM, 2017b.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pp. 1278–1286, 2014.

Salimans, T., Karpathy, A., Chen, X., and Kingma, D. P. PixelCNN++: Improving the pixelCNN with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

Samangouei, P., Kabkab, M., and Chellappa, R. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BkJ3ibb0-.

Schott, L., Rauber, J., Bethge, M., and Brendel, W. Towards the first adversarially robust neural network model on MNIST. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1EHOsC9tX.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Smith, L. and Gal, Y. Understanding measures of uncertainty for adversarial example detection. In *Uncertainty in Artificial Intelligence*, 2018.

Song, Y., Kim, T., Nowozin, S., Ermon, S., and Kushman, N. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJUYGxbCW.

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

Tabacof, P., Tavares, J., and Valle, E. Adversarial images for variational autoencoders. *arXiv preprint arXiv:1612.00155*, 2016.

Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rkZvSe-RZ.

Uesato, J., ODonoghue, B., Kohli, P., and Oord, A. Adversarial risk and the dangers of evaluating against weak attacks. In *International Conference on Machine Learning*, pp. 5032–5041, 2018.

van den Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pp. 1747–1756, 2016.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.