E-ABS: Extending the Analysis-By-Synthesis Robust Classification Model to More Complex Image Domains

An Ju
an_ju@berkeley.edu
EECS, University of California
Berkeley, CA, USA

David Wagner
daw@cs.berkeley.edu
EECS, University of California
Berkeley, CA, USA

ABSTRACT

Conditional generative models, such as Schott et al.'s Analysis-by-Synthesis (ABS), have state-of-the-art robustness on MNIST, but fail in more challenging datasets. In this paper, we present E-ABS, an improvement on ABS that achieves state-of-the-art robustness on SVHN. E-ABS gives more reliable class-conditional likelihood estimations on both in-distribution and out-of-distribution samples than ABS. Theoretically, E-ABS preserves ABS's key features for robustness; thus, we show that E-ABS has similar certified robustness as ABS. Empirically, E-ABS outperforms both ABS and adversarial training on SVHN and a traffic sign dataset, achieving state-of-the-art robustness on these two real-world tasks. Our work shows a connection between ABS-like models and some recent advances on generative models, suggesting that ABS-like models are a promising direction for defending adversarial examples.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; • **Security and privacy**;

KEYWORDS

adversarial examples, generative model, deep learning

ACM Reference Format:
An Ju and David Wagner. 2020. E-ABS: Extending the Analysis-By-Synthesis Robust Classification Model to More Complex Image Domains. In *13th ACM Workshop on Artificial Intelligence and Security (AISec'20), November 13, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3411508.3421382

1 INTRODUCTION

Deep neural networks are susceptible to adversarial examples: a deep model's accuracy drops significantly under adversarially chosen perturbations, even though these perturbations do not change human perception [7, 45]. In this paper, we want to improve adversarial robustness against small imperceptible perturbations. Specifically, our goal is to build an image classifier that maintains good accuracy under perturbations bounded by a L_p ball.

AISec'20, November 13, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-8094-2/20/11…$15.00
https://doi.org/10.1145/3411508.3421382

To this end, Schott et al. proposed Analysis-by-Synthesis (ABS) [42] and achieved state-of-the-art robustness on MNIST [28] against any L_p bounded perturbations. Golan et al. compared several defenses with human subjects and found that ABS's adversarial examples are more likely to fool human subjects [16]. Therefore, ABS opens a promising research direction on defending adversarial examples.

Despite state-of-the-art robustness on MNIST, ABS fails on more challenging datasets such as SVHN [35] and CIFAR-10 [26]. Schott et al. pointed out that ABS has low clean accuracy on CIFAR-10. Fetaya et al. found that behaviors of ABS-like models are different from MNIST on CIFAR-10 and these undesired behaviors cause underperformance on CIFAR-10. Based on their observations, Fetaya et al. claimed that ABS-like models are ineffective classifiers on complex images [15].

In this paper, we address the issues of ABS-like models with better generative models. Generative models are the building block of ABS. Given a K-class classification task, ABS learns K class-conditional data distributions with generative models. At inference time, ABS estimates the input's conditional likelihood of each class and classifies with Bayes' rule. Therefore, ABS needs high-quality conditional likelihood estimates. However, several studies suggest that generative models may give unreliable estimates to out-of-distribution samples on complex datasets [8, 19, 33]. This explains Fetaya et al.'s observation, where ABS-like models give a high likelihood to an interpolation of two images. Besides, variational autoencoders [25], the generative model used by Schott et al., could fail to learn a distribution of latent vectors that matches the prior [10, 40]; this also undermines ABS's performance.

E-ABS introduces three extensions to address these issues. First, we use adversarial autoencoders [30] to improve estimates for in-distribution samples. Second, we optimize a variational distribution at inference time. Third, we introduce a discriminative loss that uses outlier exposure [19] to improve the model's estimates for out-of-distribution samples. These extensions improve E-ABS's clean accuracy and robust accuracy on datasets that are more complex than MNIST while retaining ABS's certified robustness.

Empirically, we show that E-ABS outperforms adversarial training [29] and ABS [42] on several real-world datasets. We run extensive experiments on two simple datasets (MNIST [28] and Fashion MNIST [51]) and two more challenging real-world datasets (SVHN [35] and a dataset of European traffic signs). We measure robustness against a wide range of attacks with carefully chosen parameters. Results suggest that E-ABS preserves ABS's performance on simple datasets and sets a new state-of-the-art on SVHN and traffic signs superior to prior work.

This paper is organized as follows: Section 2 introduces ABS; Section 3 explains E-ABS's extensions; Section 4 and Section 5

present E-ABS's implementation and experiments where we compare E-ABS with other baseline defenses on four datasets. Section 6 discusses other defenses and some relevant studies on generative models. Section 7 is a discussion on some future research directions to improve E-ABS.

2 BACKGROUND

ABS classifies with class-conditional likelihood estimates. Given a datum (x, y) where $x \in \mathcal{X} \subset \mathbb{R}^N, y \in \mathcal{Y} = \{1, \ldots, K\}$, [1]

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} \propto P(x|y)P(y).$$

Accordingly, ABS esimates $P(x|k), k \in \mathcal{Y}$ and chooses the class with highest likelihood as its prediction.

Schott et al.'s ABS uses variational autoencoders (VAEs) [25, 38] for class-conditional likelihood estimation. VAEs use variational inference to estimate $P(X)$ [2]. Given a variational distribution $Q(Z)$ where $Z \in \mathcal{Z} = \mathbb{R}^M$ is a latent representation, we have a lower bound of $P(x)$ from

$$\log P(X) - \mathrm{D}_{\mathrm{KL}}[Q(Z)\|P(Z|X)] = E_{Z\sim Q(Z)}[\log P(X|Z)] - \mathrm{D}_{\mathrm{KL}}[Q(Z)\|P(Z)] \quad (1)$$

where $\mathrm{D}_{\mathrm{KL}}[\cdot\|\cdot]$ is KL-divergence. Since KL-divergence is non-negative, the right side is a lower bound for $P(X)$ known as the Evidence Lower Bound (ELBO).

The choice of $Q(Z)$ is arbitrary, but better $Q(Z)$ gives a tighter bound. VAEs use an encoder to propose a variational distribution $Q(Z|X)$ and a decoder to estimate $\log P(X|Z)$. The encoder maps an image $x \in \mathcal{X}$ to the parameters of $Q(Z|x)$; typically $Q(Z|x)$ is a multivariate Gaussian, and thus the encoder outputs a mean vector and a variance vector. The decoder maps a latent vector $z \in \mathcal{Z}$ back to the input space $\mathcal{R}^N$; the output is viewed as the mean of a Gaussian distribution, so $\log P(x|z)$ becomes $\|x - G(z)\|_2^2$ where $G(z)$ is the reconstructed image.

Given a K-class classification task, ABS trains K class-specific VAEs; the VAE for class $k \in \mathcal{Y}$ maximizes in-distribution sample likelihood by optimizing the ELBO objective (1) on $\{(x, y)|y = k\}$. At test time, as encoders are deep neural networks that are susceptible to attack, ABS replaces the encoder with an optimization step and estimates the class-conditional likelihood $\log P(x|k)$ as

$$\max_{z\in\mathcal{Z}} \left[\|x - G_k(z)\|_2^2 - \beta\, \mathrm{D}_{\mathrm{KL}}[\mathcal{N}(z, \mathbf{1})\|\, \mathcal{N}(\mathbf{0}, \mathbf{1})]\right] \quad (2)$$

where $\mathcal{N}(\mu, \Sigma)$ is a multivariate Gaussian distribution with mean μ and variance Σ, and β is a hyperparameter [20]. Denote the optimal z in (2) with z^*. Inference uses the variational distribution $\mathcal{N}(z^*, \mathbf{1})$ instead of the distribution $Q(Z|x)$ given by the encoder. It avoids encoders and thus is more robust to adversarial perturbations.

Conceptually, ABS learns K class-specific data manifolds and classifies an input x by its distance to these manifolds. Figure 1 shows intuition on why ABS's predictions are stable under small perturbations. When the learned manifolds are good representations of the real data distribution, ABS pushes adversarial examples towards the human-perceptual decision boundary [42].

[1] In this paper, we use X, Y to represent random variables, x, y to represent data, and $\mathcal{X}, \mathcal{Y}$ to represent sets.

[2] In ABS, $P(X)$ becomes $P(X|Y)$ for a class-specific VAE.

3 E-ABS DESCRIPTION

3.1 Generative Models

We use adversarial autoencoders (AAE) [30] to estimate class-conditional probabilities, because VAEs may fail to match the prior $P(z)$ and give unreliable likelihood estimates [10, 40], AAE uses a discriminator D to distinguish latent vectors encoded from input images from vectors sampled from the prior. AAE trains the encoder and discriminator like a generative adversarial network (GAN) [17], pushing the encoder's marginal distribution $Q(Z)$ to match the prior $P(Z)$ [30].

We denote the discriminator's output with $D(z)$. $D(z)$ is the probability that an input z is sampled from the prior so $0 \le D(z) \le 1$. Accordingly, the objective for training AAE's encoder and decoder is to minimize

$$\mathop{\mathbb{E}}_{(x,y)\sim P(X,Y)} \mathop{\mathbb{E}}_{z\sim Q_\phi(Z|x)} \left[c(x, G_\theta(z)) - \beta \log D_\eta(z)\right] \quad (3)$$

where θ, ϕ, η denote model parameters, $c(\cdot, \cdot)$ is a cost function such as squared error, and β is a hyperparameter. Similar to GANs, the objective for training discriminators is to minimize

$$\mathop{\mathbb{E}}_{x\sim P(X)} \mathop{\mathbb{E}}_{\substack{z\sim Q_\phi(Z|x)\\ \tilde{z}\sim P(Z)}} -\left(\log D_\eta(\tilde{z}) + \log\left(1 - D_\eta(z)\right)\right)$$

Although AAEs do not give an explicit probability estimation like VAEs, they implicitly minimize the Wasserstein distance [1, 3] between the learned $P_\theta(X)$ and the prior $P(X)$. Tolstikhin et al. showed that AAE's objective (3) is a relaxed version of the optimization

$$\inf_{Q:Q_Z=P_Z} \mathop{\mathbb{E}}_{x\sim P(X)} \mathop{\mathbb{E}}_{z\sim Q(Z|x)} [c(x, G(z))]$$

which is the Wasserstein distance between $P_\theta(X)$ and $P(X)$ under a cost c [49]. Therefore, (3) measures the distance of the input image to the learned manifold under a cost, and we use (3) for classification in E-ABS. In our experiments, we choose L_2 distance as our cost function, which means $c(x, y) = \|x - y\|_2^2$.

3.2 Discriminative Loss

A discriminative loss is used at training time to expose conditional generative models to out-of-distribution samples. Hendrycks et al. showed that outlier exposure fixes generative models' undesired behavior on out-of-distribution samples [8, 19, 33], improving the quality of likelihood estimates [19]. Specifically, we minimize E-ABS's cross-entropy with respect to class-conditional likelihood estimates. This loss facilitates each conditional AAE to recognize out-of-distribution samples and avoid giving high likelihood estimates to these samples.

The discriminative loss necessitates a structural change where class-specific encoders are replaced by a shared encoder. With class-specific encoders, a discriminative loss hinders each encoder to match the marginal distribution of latent vectors because the loss encourages a higher discriminator loss $-\log D(z)$ for OoD samples. To address this, we use the same encoder for all classes; because all samples are in-distribution for the encoder, this issue does not arise. Formally, given a datum (x, y) and $l_k(x)$ denotes the model's

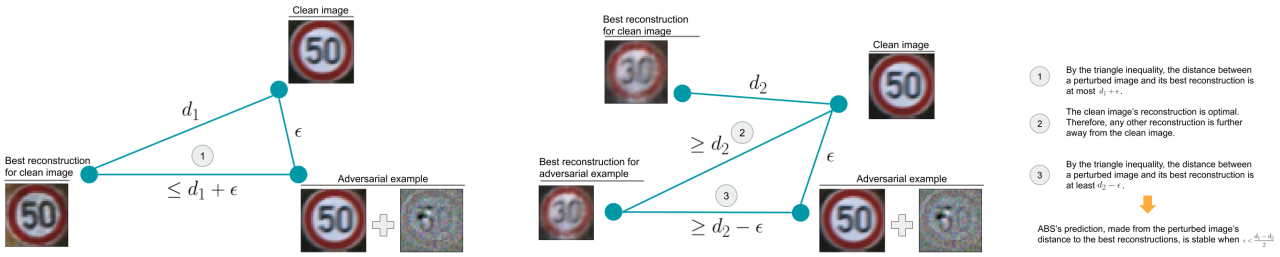


Figure 1: A simplified explanation of why ABS's predictions are stable under perturbations. Left: suppose a clean 50 image is distance d_1 away from its reconstruction using the model for class 50; then a perturbed image (of l_2 norm ϵ from the original) will be at most $d_1 + \epsilon$ away from its optimal reconstruction. Right: suppose the clean image is distance d_2 away from its reconstruction using the model for class 30; then the perturbed image will be at least $d_2 - \epsilon$ away from its optimal reconstruction. Therefore, the classification is stable when $\epsilon < (d_1 - d_2)/2$.

estimate (3) for a class $k \in \mathcal{Y}$, the discriminative loss is defined as

$$
\begin{aligned}
& -\log \frac{e^{-l_y(x)}}{\sum_{k\in\mathcal{Y}} e^{-l_k(x)}} \\
&= - \mathop{\mathbb{E}}_{z\sim Q(Z|x)} \log \frac{e^{-c(x,G_y(z))+\beta \log D(z)}}{\sum_k e^{-c(x,G_k(z))+\beta \log D(z)}} \\
&= - \mathop{\mathbb{E}}_{z\sim Q(Z|x)} \log \frac{e^{-c(x,G_y(z))}}{\sum_k e^{-c(x,G_k(z))}} \qquad (4)
\end{aligned}
$$

Because of the shared encoder, the effect of the discriminator $D(z)$ has cancelled out, so with this change to the architecture, the discriminative loss no longer encourages the encoder to produce latent vectors far away from the prior for OoD samples.

Combining (3) and (4), the training objective for encoders and decoders in E-ABS is

$$
\begin{aligned}
\mathop{\mathbb{E}}_{(x,y)\sim P(X,Y)} \mathop{\mathbb{E}}_{z\sim Q_\phi(Z|x)} & c(x, G_{\theta_y}(z)) - \beta \log D_\eta(z) \\
& - \gamma \log \frac{e^{-c(x,G_{\theta_y}(z))}}{\sum_{k\in\mathcal{Y}} e^{-c(x,G_{\theta_k}(z))}} \qquad (5)
\end{aligned}
$$

where β and γ are hyperparameters. Algorithm 1 summarizes the training method. In practice, we update discriminators and encoders/decoders in an interleaved fashion; we choose $\beta = 1$ and $\gamma = 10$ for all datasets.

3.3 Variational Inference

ABS-like models estimate the likelihood for each class through an optimization process in the latent space that maximizes the likelihood estimate. Schott et al. fix the variance of the variational distribution $Q(Z)$ during this optimization and optimize its mean. Therefore, the KL divergence term drives latent vectors towards the origin, moving away from the Gaussian prior's typical set [34]. For AAE, such an inference method leads to outlier latent vectors that significantly deviate from the prior because AAE's discriminator is a non-smooth neural network. These are undesired behaviors that undermine the model's performance.

To address this issue, we optimize both mean and variance of the variational distribution $Q^*(Z)$ at test time. Formally, we use

Algorithm 1 Training E-ABS.

Inputs: Hyperparameters $\beta > 0, \gamma > 0$.
Initialize parameters of the encoder Q_ϕ, the discriminator D_η, and K decoders $G_{\theta_1}, \ldots, G_{\theta_K}$.
repeat
  Sample $(x_1, y_1), \ldots, (x_n, y_n)$ from the training set.
  Sample $\tilde{z}_1, \ldots, \tilde{z}_n$ from the prior $P(Z)$.
  Sample z_i from $Q_\phi(Z|x_i)$ for $i = 1, \ldots, n$.
  Update ϕ and θ_i for $i = 1, \ldots, K$ by descending

$$
\begin{aligned}
& \frac{1}{n}\sum_{j=1}^{n} c(x_i, G_{\theta_{y_i}}(z_i)) - \beta \log D_\eta(z_i) \\
& \qquad - \gamma \log \frac{e^{-c(x_i,G_{\theta_{y_i}}(z_i))}}{\sum_{k=1}^{K} e^{-c(x_i,G_{\theta_k}(z_i))}}
\end{aligned}
$$

  Update η by descending

$$
-\frac{\beta}{n}\sum_{i=1}^{n} \log D_\eta(\tilde{z}_i) + \log\left(1 - D_\eta(z_i)\right)
$$

until convergence.

gradient methods to find

$$
\min_{\mu,\Sigma} \mathop{\mathbb{E}}_{z\sim\mathcal{N}(\mu,\Sigma)} [c(x, G(z)) - \beta \log D(z)] \qquad (6)
$$

The reparameterization trick [38] allows us to optimize the variational distribution's parameters μ and Σ directly.

To avoid local minima for the optimization process, Schott et al. start the optimization from the best point out of 8000 random vectors. Similarly, we sample 8000 variational distributions that are parameterized by random means and unit variance. We also include $Q_\eta(Z|x)$, the encoder's output, as a candidate. Therefore, we choose the best from 8000 + 1 variational distributions as the starting point of the optimization.

When optimizing (6), we use one sample to compute the expectation. Importance weighted sampling gives better estimations with more samples [5]. In practice, we find that a one-sample approximation is sufficient for reliable estimation and is more efficient than importance weighted sampling.

3.4 Lower Bounds for the Robustness of E-ABS

Using the same technique from Schott et al., we can deduce a lower bound for the distance to the nearest adversarial examples for E-ABS. For simplicity, we analyze L_2 bounded perturbations and use an exact z instead of a variational distribution $Q(z)$. We analyze the case where $c(x, y) = \|x - y\|_2^2$.

Given an input $x \in \mathcal{X}$ and a class $k \in \mathcal{Y}$, our estimate of $-\log P(x|k)$ is given by

$$l_k^*(x) = \min_{z \in \mathcal{Z}} \left[\|x - G_k(z)\|_2^2 - \beta \log D(z) \right].$$

Let z_k^* denote the optimal z for class k. Given a perturbation δ where $\|\delta\|_2 \leq \epsilon$, under certain conditions, we have

$$(d_k(x) - \epsilon)^2 \leq l_k^*(x + \delta) \leq l_k^*(x) + 2\epsilon \|x - G_k(z_k^*)\|_2^2 + \epsilon^2$$

where $d_k(x) = \min_{z \in \mathcal{Z}} \|x - G_k(x)\|_2$. As ABS-like models make predictions with $\arg\min_k l_k^*(x)$, adversarial perturbations increase $l_y^*(x)$ while decreasing $l_k^*(x)$, and the optimal perturbation is achieved when

$$l_y^*(x) + 2\epsilon \|x - G_y(z_y^*)\|_2^2 + \epsilon^2 = (d_k(x) - \epsilon)^2$$

for some k. Therefore, we have the following lower bound on ϵ:

$$\epsilon^* = \min_{k \in \mathcal{Y}} \frac{d_k^2(x) - l_y^*(x)}{2d_k(x) + 2\|x - G_y(z_y^*)\|_2^2} \tag{7}$$

Section A in the appendix gives more details about this bound, including its proof.

(7) suggests that robustness improves when $l_y^*(x)$ decreases and $d_k(x)$ increases, which has a direct connection to the goodness of generative models. On the one hand, $l_y^*(x)$ is lower when generative models can model in-distribution samples well. On the other hand, increasing $d_k(x)$ means that the learned manifold $\{G_k(z)|z \in \mathcal{Z}\}$ for class k is away from samples from other classes.

4 EXPERIMENTS

4.1 Datasets

We evaluate our models on four datasets. At training time, we augment datasets with additive Gaussian noise, except for adversarially trained models. We use a random 10% of each dataset's training set as a validation set.

MNIST is a dataset of handwritten digits [28]. MNIST has a clean background and binarized values, making it naturally robust against some adversarial perturbations. Schott et al. showed that binarized CNN, a simple extension to CNN models that exploits the binarized value distribution, can achieve robustness comparable with Madry et al.'s adversarially trained models [42].

Fashion MNIST is a dataset proposed by Xiao et al. as an MNIST alternative [51]. Previous studies suggest that Fashion MNIST is more challenging than MNIST. Therefore, we use Fashion MNIST to complement our comparison on simple datasets.

SuperTraffic-10 is a European traffic sign dataset composed of three datasets: the German Traffic Sign (GTS) dataset [44], the DFG traffic sign dataset [46], and the Belgium Traffic Sign (BTS) dataset [48]. We merge the three datasets because they all contain European traffic signs. We filter out small images (images that are smaller than 32×32 pixels) and choose the top 10 classes with the most images. All images in SuperTraffic-10 are 32×32 RGB images, making it a more complex dataset than MNIST or Fashion MNIST.

The Street View House Numbers (SVHN) dataset is a real-world dataset of digits [35]. SVHN is more challenging than MNIST because its images are 32×32 RGB images, and its images are taken from the real world.

4.2 Attacks

We evaluate model robustness against gradient-based attacks. ABS-like models use optimization at inference time, which makes it difficult for attacks to computing gradients. We propose an adaptive method to compute gradients. This method is an extension of Schott et al.'s Latent Descent Attack customized for ABS [42]. Furthermore, we confirm on SVHN with gradient-free attacks that E-ABS does not have obfuscated gradients [2].

All attacks are implemented with Foolbox [37].

An adaptive method to compute gradients. Adaptive attacks are necessary when evaluating model robustness [6, 50]. ABS-like models have convoluted gradients because they run several iterations of optimization at inference time. Therefore, adaptive methods to compute gradients are necessary to evaluate the robustness of these models properly.

We compute gradients from the optimal variational distribution $Q^*(Z)$ and exclude the optimization process from gradients. Specifically, given z^* sampled from the optimal $Q^*(Z)$, the reconstruction loss is the L_2 distance between x and $G(z^*)$. Therefore, its gradient is given by

$$\frac{\partial l(x, z^*)}{\partial x_i} = 2\left(x_i - G_i(z^*)\right) \tag{8}$$

where i indexes the location of a pixel in the image.

This gradient method is efficient and effective. Projected gradient descent [27, 29] using this gradient method extends and improves Schott et al.'s Latent Descent Attack [42] on ABS-like models. LDA searches for adversarial examples in the direction of the closest out-of-distribution reconstruction. A PGD attack with gradients given by (8) also pushes the adversarial example towards the best OoD reconstruction.

Unlike many other models, E-ABS's classification decision is randomized thanks to its use of the reparameterization trick at inference time. We use expectation over transformation [2], with 5 samples per batch, to deal with this randomness. Appendix B's experiments suggest that 5 samples are sufficient to stabilize gradient-based attacks.

Gradient-based attacks. We use two gradient-based L_∞ attacks:

- (**PGD**) Projected gradient descent attack [27, 29] with 80 steps. We use 5 random starts for MNIST and Fashion MNIST, and 20 random starts for SuperTraffic-10 and SVHN.
- (**DeepFool**) DeepFool attack [31] with 100 steps.

We use L_∞ PGD to choose attack parameters. In Appendix B, supplementary experiments suggest that 20 random starts and 80 steps are sufficient.

We use four gradient-based L_2 attacks:

- (**PGD**) Projected gradient descent attack with the same parameters as L_∞ PGD.

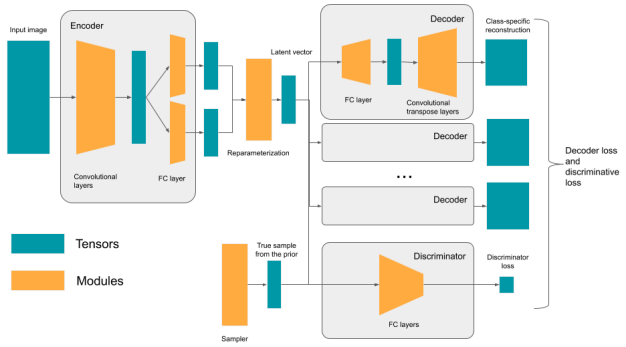


Figure 2: E-ABS's structure. E-ABS has one shared encoder with multiple decoders. The discriminator distinguishes vectors sampled from $Q(Z|X)$ from vectors sampled from $P(Z)$.

- (**DDN**) Decoupled direction and norm attack [39] with 80 steps. DDN is an extension of the PGD attack.
- (**CW**) Carlini-Wagner attack [7] with 5 binary searches and 100 steps.
- (**DeepFool**) DeepFool attack with the same parameters as L_∞ DeepFool.

Furthermore, we include an attack that adds Gaussian noise with increasingly large standard deviation. Previous studies suggest that a model's performance under additive Gaussian noises is correlated with a model's robustness against L_2 attacks [14].

Gradient-free attacks. We use two gradient-free attacks: a Boundary attack [4] and a PGD attack with gradient estimation. We use 100000 iterations for the Boundary attacks and 50 steps for the PGD attack with gradient estimation. We initialize the Boundary attack with a random sample that is classified as a different class from the ground truth.

4.3 Models

We run attacks against E-ABS and two baseline defenses: Schott et al.'s ABS model and an adversarially trained model. Our E-ABS model uses a convolutional encoder and decoder; the discriminator is a two-layer feed-forward network. Figure 2 shows E-ABS's structure. The other models share E-ABS's modules. ABS uses the same model structure as E-ABS, but does not have discriminators. CNNs add a linear layer after E-ABS's encoder for classification. Adversarially trained CNNs are tuned to have comparable clean data accuracy with E-ABS. Because adversarially trained CNNs under L_∞ attacks do not generalize to L_2 attacks [42], we include both Adv-L_∞, an adversarially trained model with L_∞ attacks, and Adv-L_2, an adversarially trained model with L_2 attacks.

In our experiments, the E-ABS encoder uses 4 convolutional layers followed by two parallel fully-connected layers that compute the variational distribution's parameters. All convolutional layers use batch normalization [21] and Leaky ReLU activation [52]. We use dropout [43] after the last convolutional layer. The decoder uses a fully-connected layer followed by convolution transpose layers. The fully-connected layer's output has the same size as the first convolution transpose layer's output depth. We use dropout [43] after the fully-connected layer. All convolution transpose layers, except the last layer, use batch normalization [21] and ReLU activation [32]; the last layer uses sigmoid. Table 9 in Appendix C shows more details of the architecture we use in our experiments.

We use Adam [23] and batch size 512 to train all models. We use an initial learning rate 0.001 for CNN, ABS, and E-ABS, and halve the learning rate every 200 epochs. We train CNN, ABS, and E-ABS for 500 epochs, except for ABS and E-ABS on SVHN, where we train 800 epochs. For the adversarially trained model, we use the pre-trained CNN model and retrain the model for 200 epochs with a learning rate 0.0001 and no learning rate decay.

At training time, we augment data with additive Gaussian noise. We use a standard deviation 0.2 for MNIST and Fashion MNIST, and 0.01 for SuperTraffic-10 and SVHN.

4.4 Ablation Study

E-ABS consists of three separate extensions to ABS. We present an ablation study to demonstrate that all three extensions are necessary. We denote the three extensions with A (for AAE-based models), D (for models with a discriminative loss and shared encoder), and V (for models that use variational inference). This ablation study examines all combinations of the three extensions on SVHN with L_∞ and L_2 PGD attacks.

When training A-ABS and AV-ABS models, we find that training discriminators with both in-distribution samples and out-of-distribution samples improves the model's performance. This way, class-specific discriminators see latent vectors encoded from not only in-distribution images but also out-of-distribution images.

4.5 Metrics

We report each model's clean accuracy and accuracy under bounded perturbations. Unless otherwise specified, we choose 1000 random test samples when reporting robustness results. We run McNemar's test [12] on every pair of models to test whether the difference in their performance is statistically significant; McNemar's test is a pairwise test with a null hypothesis that none of the models performs better than the other. We choose $\alpha = 0.01$. Besides results under each attack, we report the model's accuracy under the combination of all attacks of the same type.

5 RESULTS

E-ABS extends ABS to more complex image datasets.

E-ABS outperforms ABS on SVHN and SuperTraffic-10, as shown in Table 3 and Table 4. For example, ABS's clean data accuracy on SVHN is only 45.8%; E-ABS increases clean accuracy to 89.2%, comparable with an unprotected CNN (91.3%). This suggests that E-ABS provides better likelihood estimates on image datasets that were previously considered challenging for ABS-like models [15].

E-ABS also outperforms adversarial training on SuperTraffic-10 and SVHN and achieves a new state of the art in robustness on these datasets.[3] Also, E-ABS is robust against both L_∞ and L_2 attacks, while adversarially trained models have robustness only

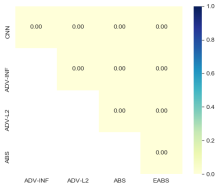
[3]The best-published result on SVHN that we know of is 55.59% accuracy under PGD-L_∞ attack with $\epsilon = 8/255$ [18]; E-ABS achieves 57%. We create SuperTraffic-10, but GTSRB [44], one of the three datasets included in SuperTraffic-10's, has the best-known result of 67.9% under PGD-L_2 attack with $\epsilon = 0.2$ [9], according to robust-ml.org. E-ABS's accuracy is 85% under the same attack.

Table 1: Results of different models on MNIST. Reported numbers are accuracy under bounded perturbations. Results are based on 1000 samples. McNemar's test shows that all differences are significant ($p < 0.01$).

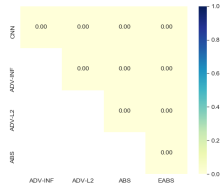
	CNN	Adv-L_∞	Adv-L_2	ABS	E-ABS
Clean	99.5%	98.5%	98.6%	96.1%	99.4%
L_∞ attack ($\epsilon = 0.3$)					
PGD	0%	90.4%	0%	3.4%	17.1%
DeepFool	17.4%	92.9%	43.8%	6.2%	31%
Noise	99.3%	98.4%	98.5%	95.8%	99.1%
All L_∞	0%	90.3%	0%	3.3%	16.5%
L_2 attack ($\epsilon = 1.5$)					
PGD	72.8%	88.4%	89.3%	75.2%	90.6%
DDN	62.3%	81.7%	87.5%	74.1%	91.4%
CW	86.8%	91.2%	94.7%	82.8%	94.8%
DeepFool	83.8%	93.1%	92.2%	95.9%	91.1
Noise	99.3%	98.5%	98.6%	74.4%	99.4%
All L_2	61.8%	81.5%	87.5%	73.7%	90.4%

McNemar's test p values

L_∞ attacks



L_2 attacks



against the type of attack used at training time. Figure 3 show the model's accuracy under PGD attacks as a function of the size of the perturbation on SVHN. It confirms that E-ABS outperforms other baselines for both L_∞ and L_2 attacks. Furthermore, all models lose accuracy under large perturbations, suggesting that our adaptive gradient methods are correct.

On MNIST and Fashion MNIST, E-ABS's clean accuracy is comparable with unprotected CNNs, and its robustness is comparable to ABS and adversarial training, as shown in Table 1 and Table 2. Both ABS and E-ABS lose robustness under L_∞ attacks with a large bound, as shown in Table 1. We believe this occurs because likelihood estimates rely on the L_2 distance between the input and its reconstruction. It may be possible to use other distance metrics to target L_∞ robustness more specifically.

On MNIST, our results are mostly consistent with [42]. Schott et al.'s ABS has better clean data accuracy than ours. In comparison, our ABS baseline uses the same structure as E-ABS, with more capacity and dropout layers. Our experiments suggest that these structural differences could explain most of the difference. Our unprotected CNN models have better robustness than Schott et al. observed in their experiments because we use Gaussian noise to augment data at training time.

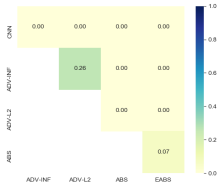
Results with gradient-free attacks confirm that our model does not have obfuscated gradients. E-ABS has 92.0% accuracy under

Table 2: Results of different models on Fashion MNIST. Reported numbers are accuracy under bounded perturbations. Results are based on 1000 samples. McNemar's test shows that all differences are significant ($p < 0.01$) except for: Adv-L_∞ and Adv-L_2 under L_∞ attacks, ABS and E-ABS under L_∞ attacks, Adv-L_∞ and ABS under L_2 attacks, and ABS and E-ABS under L_2 attacks.

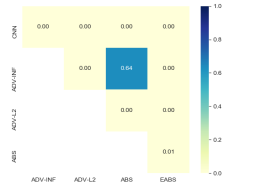
	CNN	Adv-L_∞	Adv-L_2	ABS	E-ABS
Clean	91.0%	88.9%	87.4%	81.6%	90.1%
L_∞ attack ($\epsilon = 0.1$)					
PGD	9.4%	57%	56%	46.2%	43.5%
DeepFool	25.9%	64.9%	63.2%	47.7%	45.7%
Noise	90.5%	88.9%	86.5%	81.4%	0.9%
All L_∞	9.4%	57%	55.7%	45.8%	43.5%
L_2 attack ($\epsilon = 1.5$)					
PGD	19.1%	51.9%	59.2%	46.6%	43.3%
DDN	13.4%	46.6%	57.2%	45.4%	49.9%
CW	30.7%	61%	63.6%	50.7%	54.1%
DeepFool	28.9%	56.4%	60.4%	46.4%	44.3%
Noise	90.9%	88.8%	87.1%	81.2%	89.6%
All L_2	13.2%	45.9%	55.1%	45.2%	41.7%

McNemar's test p values

L_∞ attacks



L_2 attacks



Boundary attack and 88.0% accuracy under L_∞ PGD attack with gradient estimators. Comparing with Table 4, gradient-based attacks with the adaptive gradient method are much stronger than gradient-free attacks.

On MNIST, E-ABS's adversarial perturbations are semantically consistent with human perception, similar to ABS [16, 42]. Figure 4 shows some adversarial examples for E-ABS on MNIST under a PGD-L_2 attack. However, on more complex datasets such as SVHN, E-ABS does not match human perception, as shown in Figure 5.

In summary, results from Table 1 to Table 4 suggest that

- E-ABS has similar or better clean data accuracy than ABS on both complex and simple datasets.
- Compared with ABS, E-ABS has comparable robustness on simple datasets and better robustness on complex datasets.
- E-ABS outperforms other models on SuperTraffic-10 and SVHN, providing a new state-of-the-art on these datasets.
- E-ABS's clean data accuracy is comparable with an unprotected CNN on all datasets except for SuperTraffic-10.

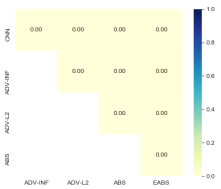
All three extensions are necessary for E-ABS.

Table 3: Results of different models on SuperTraffic-10. Reported numbers are accuracy under bounded perturbations. Results are based on 1000 samples. McNemar's test shows that all differences are significant ($p < 0.01$) except for Adv-L_∞ and ABS under L_2 attacks.

	CNN	Adv-L_∞	Adv-L_2	ABS	E-ABS
Clean	99%	91.4%	91.6%	84.9%	92.7%
L_∞ attack ($\epsilon = 8/255$)					
PGD	29.8%	74.3%	60.2%	53.4%	70.9%
DeepFool	48.1%	74.9%	64.6%	54.5%	82.1%
Noise	99.1%	91.7%	91.6%	84.8%	91.7%
All L_∞	29.8%	73.7%	59.9%	53.2%	69.8%
L_2 attack ($\epsilon = 1.5$)					
PGD	14.8%	46.2%	53.7%	48%	66.1%
DDN	9.3%	44.6%	52.4%	47.1%	73%
CW	14.5%	51%	54.5%	47%	73.8%
DeepFool	33%	49%	56.6%	48.7%	93.3%
Noise	98.6%	91.6%	91.4%	86.3%	73.8%
All L_2	8.7%	44.3%	52.2%	46.3%	59.9%

McNemar's test p values

L_∞ attacks



L_2 attacks

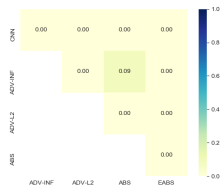
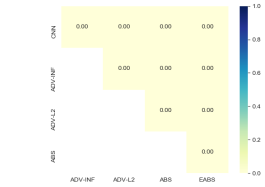


Table 4: Results of different models on SVHN. Reported numbers are accuracy under bounded perturbations. Results are based on 1000 samples. McNemar's test shows that all differences are significant ($p < 0.01$) except for Adv-L_∞ and ABS under L_∞ attacks.

	CNN	Adv-L_∞	Adv-L_2	ABS	E-ABS
Clean	91.3%	89.0%	87.1%	45.8%	89.2%
L_∞ attack ($\epsilon = 8/255$)					
PGD	0%	37.2%	26.7%	6.6%	57%
DeepFool	0.1%	50.4%	36.6%	8.4%	64.1%
Noise	90.7%	89%	87%	45.7%	89.3%
All L_∞	0%	37.2%	26.7%	6.6%	55.7%
L_2 attack ($\epsilon = 1.5$)					
PGD	0%	5.1%	13.4%	4.4%	40.9%
DDN	0%	3.7%	11.6%	4.3%	57.8%
CW	0%	5.5%	13.7%	4.5%	47.6%
DeepFool	0%	13.4%	19.1%	4.7%	51.6%
Noise	90%	89.4%	87.4%	43.5%	88.8%
All L_2	0%	3.6%	10.9%	4.3%	36%

McNemar's test p values

L_∞ attacks



L_2 attacks

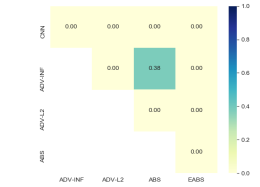


Table 5 compares all combinations of Section 3's three extensions. Table 5 suggests that no single technique can significantly improve ABS's performance. AV-ABS and DV-ABS outperform ABS on clean accuracy, suggesting that variational inference improves the model's accuracy. However, variational inference alone gives no improvement (V-ABS). On the other hand, exact inference significantly undermines clean accuracy on models with adversarial autoencoders; both A-ABS and AD-ABS have a worse accuracy than ABS. We hypothesize that in these cases, exact inference leads to latent vectors away from the Gaussian prior's typical set [34].

6 RELATED WORK

6.1 Defend against perturbations

Many defenses have been proposed since Szegedy et al. found that deep neural networks are susceptible to carefully engineered adversarial inputs [45]. However, later studies showed that many of these defenses are breakable [7]. Among these defenses, two have shown promising results and have withstood tests so far.

One is adversarial training [29]. This line of research focuses on methods that use carefully designed training examples to improve model robustness. However, it only provides robustness under a specific perturbation type and may not generalize to other perturbations.

Another defense that is believed to be useful is Schott et al.'s ABS model [42], which we described in Section 1. Others have proposed other defenses that use generative models [41], but those defenses are broken under more powerful attacks [22]. Unlike those models, ABS uses class-conditional probabilities estimated by generative models. ABS's decision-making process does not involve functions that map from a high-dimensional space to a low-dimensional space, which might explain why ABS is robust against adversarial examples.

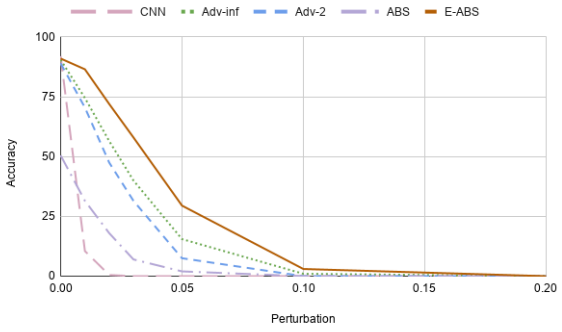
6.2 Generative models for likelihood estimation

Variational autoencoders [25, 38] give a lower bound for the likelihood of a sample, as introduced in Section 1. Generative flows [13, 24] could compute the exact likelihood of an input. Compared with VAEs, generative flows are computationally expensive as they keep the input dimension unchanged. A recent study [36] shows that generative flows are susceptible to adversarial attacks.

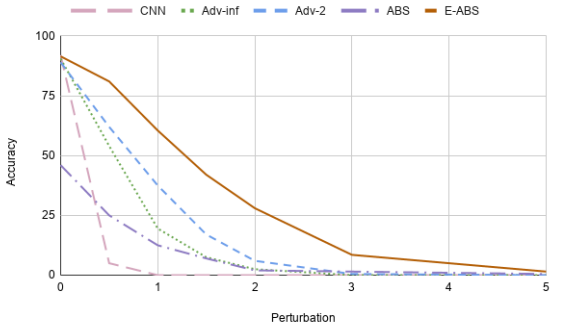
Generative adversarial networks (GANs) [17] are strong generative models, achieving better reconstructions than VAEs in many

Table 5: An ablation study on SVHN. We report model accuracy under bounded PGD attacks.

	ABS	A-ABS	D-ABS	V-ABS	AD-ABS	AV-ABS	DV-ABS	ADV-ABS (E-ABS)
CLEAN	45.8%	23.7%	44.9%	44.1%	30.8%	60.1%	65.7%	89.2%
PGD-L_∞	6.6%	6.4%	8.4%	2.5%	2.3%	11%	18.9%	57%
PGD-L_2	4.4%	5.9%	5.6%	2.1%	1.5%	8.2%	14%	40.9%



(a) L_∞ PGD



(b) L_2 PGD

Figure 3: Model accuracy under PGD attack, as a function of perturbation size, on SVHN.

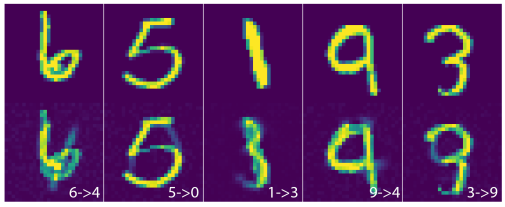


Figure 4: E-ABS's adversarial examples on MNIST under PGD-L_2 attack. Top: Clean images. Bottom: Adversarial examples.

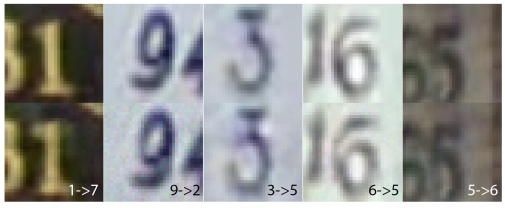


Figure 5: E-ABS's adversarial examples on SVHN under PGD-L_2 attack. Top: Clean images. Bottom: Adversarial examples.

domains. However, it remains an open question on how to obtain likelihood estimates from a GAN. Also, Theis et al. show that the quality of reconstructions and likelihood estimations are unrelated [47]. Therefore, it is unclear how to use GANs in an ABS-like model or whether this would yield any improvement.

Recently, researchers found that generative models give high probability estimates to out-of-distribution samples [33], which may explain why ABS struggles with datasets other than MNIST. Several mechanisms have been proposed since then. Hendrycks et al. [19] expose the generative model to proxy OoD samples. Their scheme is simple and applicable to ABS-like models. Another method uses Watanabe Akaike Information Criterion (WAIC) to address this issue [8]. Although empirically effective, the authors acknowledge that this method does not prevent the issue in theory. Furthermore, this method requires multiple models, which is costly at inference time.

7 DISCUSSION

This paper presents E-ABS, an extension to Schott et al.'s ABS model. E-ABS has state-of-the-art robustness on SVHN and a traffic sign dataset, beating Madry et al.'s adversarial training [29]. Compared with ABS, E-ABS achieves significantly better clean accuracy, comparable with unprotected CNN models on the two real-world datasets. Therefore, E-ABS has successfully addressed the shortcomings of ABS observed by Fetaya et al. [15], suggesting that robust classification with conditional generative models is a promising research direction.

Despite the improvements, E-ABS still has limitations on more complex datasets. On CIFAR-10 [26], E-ABS achieves 60% clean accuracy, leaving a significant gap to state-of-the-art convolutional models. Therefore, there are still obstacles to use conditional generative models in datasets such as CIFAR and ImageNet [11]. Two obstacles are particularly critical.

First, generative models are sensitive to image similarity measures. We use L_2 distance to measure the image similarity. However, this metric is known to perform poorly in high-dimensional spaces: on the one hand, a one-pixel translation could lead to a large L_2 distance; on the other hand, two objects sharing the same background could be very close in terms of L_2 distance. Therefore, finding a better distance metric that captures semantic similarity while staying robust against small perturbations might further improve the performance of ABS-like models.

Second, ABS-like models have two efficiency bottlenecks. The time for inference scales linearly with the number of classes, making inference inefficient on large datasets such as CIFAR-100 and ImageNet. Also, running time grows approximately linearly with the number of iterations of optimization used during inference. With more iterations, the model is more stable and has better accuracy and robustness. Better sampling or optimization methods may allow more efficient inference.

Our work may be of independent interest as an application of generative models that give explicit likelihood estimates. Traditionally, these models yield lower-quality reconstructions than GANs, which do not give likelihood estimates. Therefore, the success of E-ABS on complex image domains motivates research into better generative models for distribution matching: progress on such models may lead to more robust models.

ACKNOWLEDGMENTS

This work was supposed by the Hewlett Foundation through the Center for Long-Term Cybersecurity, by the Berkeley Deep Drive project, and by a generous gift from Futurewei.

The authors thank Howard Ki, Yunqi (Evelyn) Li, Sachit Shroff, Zachary Golan-Strieb, and Benson Yuan for creating the SuperTraffic-10 dataset.

REFERENCES

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein GAN. *arXiv preprint arXiv:1701.07875* (2017).
[2] Anish Athalye, Nicholas Carlini, and David Wagner. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420* (2018).
[3] Olivier Bousquet, Sylvain Gelly, Ilya Tolstikhin, Carl-Johann Simon-Gabriel, and Bernhard Schoelkopf. 2017. From optimal transport to generative modeling: the VEGAN cookbook. *arXiv preprint arXiv:1705.07642* (2017).
[4] Wieland Brendel, Jonas Rauber, and Matthias Bethge. 2017. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248* (2017).
[5] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. 2015. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519* (2015).
[6] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. 2019. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705* (2019).
[7] N. Carlini and D. Wagner. 2017. Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE Symposium on Security and Privacy (SP)*. 39–57.
[8] Hyunsun Choi, Eric Jang, and Alexander A Alemi. 2018. WAIC, but why? Generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392* (2018).
[9] Francesco Croce, Maksym Andriushchenko, and Matthias Hein. 2018. Provable robustness of ReLu networks via maximization of linear regions. *arXiv preprint arXiv:1810.07481* (2018).
[10] Bin Dai and David Wipf. 2019. Diagnosing and enhancing VAE models. *arXiv preprint arXiv:1903.05789* (2019).
[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.
[12] Thomas G Dietterich. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation* 10, 7 (1998), 1895–1923.
[13] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. 2016. Density estimation using Real NVP. *arXiv preprint arXiv:1605.08803* (2016).
[14] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. 2016. Robustness of classifiers: from adversarial to random noise. In *Advances in Neural Information Processing Systems*. 1632–1640.
[15] Ethan Fetaya, Jörn-Henrik Jacobsen, Will Grathwohl, and Richard Zemel. 2019. Understanding the limitations of conditional generative models. *arXiv preprint arXiv:1906.01171* (2019).
[16] Tal Golan, Prashant C Raju, and Nikolaus Kriegeskorte. 2019. Controversial stimuli: pitting neural networks against each other as models of human recognition. *arXiv preprint arXiv:1911.09288* (2019).
[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.
[18] Minghao Guo, Yuzhe Yang, Rui Xu, and Ziwei Liu. 2019. When NAS Meets Robustness: In Search of Robust Architectures against Adversarial Attacks. *arXiv preprint arXiv:1911.10695* (2019).
[19] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. 2018. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606* (2018).
[20] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning basic visual concepts with a constrained variational framework. *ICLR* 2, 5 (2017), 6.
[21] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).
[22] Ajil Jalal, Andrew Ilyas, Constantinos Daskalakis, and Alexandros G Dimakis. 2017. The robust manifold defense: Adversarial training using generative models. *arXiv preprint arXiv:1712.09196* (2017).
[23] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[24] Durk P Kingma and Prafulla Dhariwal. 2018. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*. 10215–10224.
[25] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114* (2013).
[26] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).
[27] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533* (2016).
[28] Yann LeCun and Corinna Cortes. 2010. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/. (2010). http://yann.lecun.com/exdb/mnist/
[29] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* (2017).
[30] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. 2015. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644* (2015).
[31] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2574–2582.
[32] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*. 807–814.
[33] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. 2018. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136* (2018).
[34] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, and Balaji Lakshminarayanan. 2019. Detecting out-of-distribution inputs to deep generative models using a test for typicality. *arXiv preprint arXiv:1906.02994* (2019).
[35] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. 2011. Reading digits in natural images with unsupervised feature learning. (2011).
[36] Phillip Pope, Yogesh Balaji, and Soheil Feizi. 2019. Adversarial robustness of flow-based generative models. *arXiv preprint arXiv:1911.08654* (2019).
[37] Jonas Rauber, Wieland Brendel, and Matthias Bethge. 2017. Foolbox: A Python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131* (2017). arXiv:1707.04131 http://arxiv.org/abs/1707.04131
[38] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082* (2014).
[39] Jérôme Rony, Luiz G Hafemann, Luiz S Oliveira, Ismail Ben Ayed, Robert Sabourin, and Eric Granger. 2019. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4322–4330.
[40] Mihaela Rosca, Balaji Lakshminarayanan, and Shakir Mohamed. 2018. Distribution matching in variational inference. *arXiv preprint arXiv:1802.06847* (2018).

[41] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. 2018. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605* (2018).
[42] Lukas Schott, Jonas Rauber, Matthias Bethge, and Wieland Brendel. 2018. Towards the first adversarially robust neural network model on MNIST. *arXiv preprint arXiv:1805.09190* (2018).
[43] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15 (2014), 1929–1958. http://jmlr.org/papers/v15/srivastava14a.html
[44] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. 2012. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks* 32 (2012), 323–332.
[45] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).
[46] Domen Tabernik and Danijel Skočaj. 2019. Deep Learning for Large-Scale Traffic-Sign Detection and Recognition. *IEEE Transactions on Intelligent Transportation Systems* (2019).
[47] Lucas Theis, Aäron van den Oord, and Matthias Bethge. 2015. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844* (2015).
[48] Radu Timofte, Karel Zimmermann, and Luc Van Gool. 2014. Multi-view traffic sign detection, recognition, and 3D localisation. *Machine vision and applications* 25, 3 (2014), 633–647.
[49] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. 2017. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558* (2017).
[50] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. 2020. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347* (2020).
[51] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv preprint arXiv:1708.07747* (2017).
[52] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. 2015. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853* (2015).

A PROOFS

Our analysis is similar to Schott et al.'s analysis under L_2 attacks [42]. The main difference is that the discriminator loss $-\beta \log D(z)$ is non-negative, which simplifies our conclusions.

A.1 Assumptions and Notations

For simplicity, we study exact inference and L_2 attacks. Exact inference means that we find

$$z^* = \arg\min \|x - G(z)\|_2^2 - \beta \log D(z)$$

where $D(z)$ is discriminator's output and $0 \leq D(z) \leq 1$; β is a hyperparameter and $\beta > 0$.

A.2 A Bound of the Distance of Adversarial Examples for E-ABS

Given an input x and any class k, E-ABS estimates the class-conditional negative log-likelihood with

$$l_k^*(x) = \min_z \|x - G_k(z)\|_2^2 - \log D(z) \tag{9}$$

Given a perturbation δ where $\|\delta\|_2 = \epsilon$, we want to find a lower bound of ϵ for δ to change E-ABS's prediction from y, the ground truth, to k, a specific class $k \neq y$.

First, we show that

Claim 1.

$$l_y^*(x+\delta) \leq l_y^*(x) + 2\epsilon\|x - G_y(z^*)\|_2 + \epsilon^2$$

where $z^* = \arg\min_z \|x - G_y(z)\|_2^2 - \log D(z)$.

Proof.

$$\begin{aligned}
l_y^*(x+\delta) =& \min_z \|x+\delta - G_y(z)\|_2^2 - \beta \log D(z) \\
\leq& \|x+\delta - G_y(z^*)\|_2^2 - \beta \log D(z^*) \\
=& \|x - G_y(z^*)\|_2^2 + 2\delta^T(x - G_y(z^*)) + \epsilon^2 - \beta \log D(z^*) \\
\leq& \|x - G_y(z^*)\|_2^2 + 2\epsilon\|x - G_y(z^*)\|_2 + \epsilon^2 - \beta \log D(z^*) \\
=& l_y^*(x) + 2\epsilon\|x - G_y(z^*)\|_2 + \epsilon^2
\end{aligned}$$

□

Second, we show that

Claim 2. *Let* $d_k(x) = \min_z \|x - G_k(z)\|_2^2$ *and assume* $\epsilon < d_k(x)$,

$$l_k^*(x+\delta) \geq (d_k(x) - \epsilon)^2$$

Proof.

$$l_k^*(x+\delta) = \min_z \|x+\delta - G_k(z)\|_2^2 - \log D(z)$$

Since $0 \leq D(z) \leq 1$ and $\beta > 0$, we know

$$\begin{aligned}
l_k^*(x+\delta) \geq& \min_z \|x+\delta - G_k(z)\|_2^2 \\
=& \min_z \|x - G_k(z)\|_2^2 + 2\delta^T(x - G_k(z)) + \epsilon^2 \\
\geq& \min_z \|x - G_k(z)\|_2^2 - 2\epsilon\|x - G_k(z)\|_2 + \epsilon^2
\end{aligned}$$

By the definition of $d_k(x)$, we know

$$\|x - G_k(z)\| \geq d_k(x).$$

Therefore, when $d_k(x) > \epsilon$,

$$\begin{aligned}
l_k^*(x+\delta) \geq& d_k^2(x) - 2\epsilon d_k(x) + \epsilon^2 \\
=& (d_k(x) - \epsilon)^2
\end{aligned}$$

□

We know from the proof that when $\epsilon > d_k(x)$, $l_k^*(x+\delta) \geq 0$. With Claim 1 and Claim 2, we can find a lower bound for ϵ from

$$l_y^*(x) + 2\epsilon\|x - G_y(z^*)\|_2 + \epsilon^2 = (d_k(x) - \epsilon)^2$$

and the lower bound for an untargeted attack is the minimal of all bounds, which gives

$$\epsilon^* = \arg\min_k \frac{d_k^2(x) - l_y^*(x)}{2d_k(x) + 2\|x - G_y(z^*)\|_2}$$

This bound holds when $\epsilon < d_k(x)$, that is, adversarial perturbations are not large enough to move x to its best reconstruction $G_k(z^*)$ for some class $k \neq y$.

An interesting observation is that Claim 1 reaches the optimal value when δ has the opposite direction as $x - G_y(x)$ and Claim 2 reaches the optimal value when δ has the same direction as $x - G_k(x)$. This suggests that our gradient-based attacks that use

$$\frac{\partial l(x, z^*)}{\partial x_i} = 2 * \left(x_i - G_{ij}(z^*)\right)$$

to compute gradients are consistent with the theoretical analysis presented in this section. Schott et al.'s LDA attack is closely related to this observation as well, as the LDA attack searches adversarial examples along the direction $G_k(x) - x$.

B SUPPLEMENTARY EXPERIMENTS ON GRADIENT-BASED ATTACKS

In this section, we present some supplementary expeirments to justify our choice of parameters for gradient-based attacks. All experiments use E-ABS model.

B.1 Expectation-over-transformations

We test a range of runs for EOT. We evaluate the model's accuracy under a L_∞ PGD attack with 5 random starts on SVHN. Table 6 presents experiment results.

Table 6: Experiments on the number of runs for EOT, based on an L_∞ PGD attack 5 random starts on SVHN.

Number of runs	1	2	3	5	10
Accuracy	57.5	58.4	57.8	58.7	58.6

Table 6 suggests that 5 runs are enough to stabilize model outputs. Specifically, with fewer runs, the model has seemingly better robustness because it may misclassify due to inference-time randomness, which leads an attack to believe that it has succeeded falsely.

B.2 Number of PGD random starts

We run experiments to decide the number of random starts necessary for PGD attacks. We evaluate the model's accuracy under a L_∞ PGD attack with 100 steps. We use 3 runs for EOT for efficiency. Table 7 presents these experiments.

Table 7: Experiments on the number of random starts for PGD, based on an L_∞ PGD attack on SVHN.

Number of random starts	5	10	15	20	50
Accuracy	57.9	57.3	56.8	56	56

Based on Table 7, we choose 20 random starts for our PGD attacks on SVHN and SuperTraffic-10.

B.3 PGD steps

We evaluate the model's robustness under L_∞ PGD attacks to choose a proper number of PGD steps. We use 20 random starts for all PGD attacks, 3 runs for EOT, and 200 random samples from SVHN. We use less EOT runs and random samples for efficiency concerns. Table 8 shows the results.

Table 8: Experiments on the number of PGD steps, based on L_∞ PGD attacks on SVHN.

Number of steps	0	10	30	50	100	200	500
Accuracy (%)	87.5	78	63.5	51.5	51.5	52	51.5

Table 8 suggests that PGD attacks achieve the best results with 50 steps. We choose 80 steps in our experiments.

C MODELS AND EXPERIMENTS

Table 9 shows parameters for E-ABS. The other models (CNN, Adv-L_∞, Adv-L_2, and ABS) share the same modules.

Table 9: The model parameters for each dataset.

MNIST and Fashion MNIST								Latent dimensions: 10
Encoder				Decoder				Discriminator
Channels	Kernel	Stride	Padding	Channels	Kernel	Stride	Padding	Neurons
16	5	2	2	32	4	1	0	256
32	5	2	2	32	5	2	0	128
64	5	2	2	16	5	2	0	
64	4	1	1	8	4	1	0	
				1	1	1		

SuperTraffic-10								Latent dimensions: 16
Encoder				Decoder				Discriminator
Channels	Kernel	Stride	Padding	Channels	Kernel	Stride	Padding	Neurons
16	5	2	2	64	4	1	0	256
32	5	2	2	64	4	2	1	128
64	5	2	2	32	4	2	1	
128	4	1	0	16	4	2	1	
				3	1	1	0	

SVHN								Latent dimensions: 40
Encoder				Decoder				Discriminator
Channels	Kernel	Stride	Padding	Channels	Kernel	Stride	Padding	Neurons
32	5	2	2	64	4	1	0	512
64	5	2	2	64	4	2	1	256
128	5	2	2	32	4	2	1	
128	4	1	0	16	4	2	1	
				3	1	1	0	