

The Robust Manifold Defense: Adversarial Training using Generative Models

Ajil Jalal*
ajiljalal@utexas.edu
UT Austin

Andrew Ilyas*
ailyas@mit.edu
MIT EECS

Constantinos Daskalakis
costis@csail.mit.edu
MIT EECS

Alexandros G. Dimakis
dimakis@austin.utexas.edu
UT Austin

July 11, 2019

Abstract

We propose a new type of attack for finding adversarial examples for image classifiers. Our method exploits spanners, i.e. deep neural networks whose input space is low-dimensional and whose output range approximates the set of images of interest. Spanners may be generators of GANs or decoders of VAEs. The key idea in our attack is to search over latent code pairs to find ones that generate nearby images with different classifier outputs. We argue that our attack is stronger than searching over perturbations of real images. Moreover, we show that our stronger attack can be used to reduce the accuracy of Defense-GAN to 3%, resolving an open problem from the well-known paper by Athalye et al. We combine our attack with normal adversarial training to obtain the most robust known MNIST classifier, significantly improving the state of the art against PGD attacks. Our formulation involves solving a min-max problem, where the min player sets the parameters of the classifier and the max player is running our attack, and is thus searching for adversarial examples in the *low-dimensional* input space of the spanner.¹

1 Introduction

Deep neural network (DNN) classifiers are demonstrating excellent performance in various computer vision tasks. These models work well on benign inputs but recent work has shown that it is possible to make very small changes to an input image and drastically fool state-of-the-art models [Szegedy et al., 2013, Goodfellow et al., 2014b]. These *adversarial examples* are barely perceivable by humans, can be targeted to create desired labels even with black-box access to classifiers, and can be implemented as objects in the physical world [Papernot et al., 2016a, Kurakin et al., 2016, Athalye et al., 2017].

In this work, we propose a novel optimization perspective towards obtaining stronger attacks and stronger defenses. Our starting point is the assumption that we have access to a “spanner” for the type of data that we are considering. A spanner is a DNN $G : \mathbb{R}^k \rightarrow \mathbb{R}^n$, whose input space is low-dimensional and whose output range $\{G(z)\}_{z \in \mathbb{R}^k}$ is a good approximation to the dataset of

*Equal Contribution

¹All code and models are available at <https://github.com/ajiljalal/manifold-defense.git>

interest. A spanner could be the generator of a GAN or the decoder of an auto-encoder, which are trained on the same dataset that the classifier being defended or attacked was trained on. We are interested in answering the following question:

Can we use a spanner to obtain improved defenses or attacks of DNN based classifiers?

To answer this question, we first propose a preliminary defense approach, called Invert-and-Classify (INC), which simply uses the spanner as a “denoiser.” Given an image x we find its projection $G(z^*)$ to the range of the spanner,² and then apply the classifier C , which is being defended, on the projection $G(z^*)$, i.e. we output $C(G(z^*))$. We show that this attack is not robust by proposing our “overpowered latent-space attack,” which successfully circumvents this defense.

We note that a one-sided variant of our overpowered attack has been used by Athalye et al. [2018] to partially circumvent the DefenseGAN defense [Samangouei et al., 2018], which amounts precisely to the INC defense described above, when the spanner is instantiated as a generator from a GAN.

Our overpowered attack is formulated as a min-max optimization problem, where the adversary has more power than usual: rather than perturbing a single image x , we attempt to identify a pair of latent codes z, z' such that the images generated from these codes are similar, i.e. $\|G(z) - G(z')\|$ is small, but the classifier outputs on these images are far, i.e. $\mathcal{L}(C(G(z)), C(G(z')))$ is large, where \mathcal{L} is some loss function such as cross-entropy. We show how to use our overpowered attack to fully circumvent DefenseGAN under the settings of Athalye et al. [2018] and Samangouei et al. [2018].³ This resolves an open challenge⁴ by Athalye et al. [2018].

We also show how our overpowered attack can be used to strengthen adversarial training. We set up a min-max problem in which the min player chooses the parameters of the classifier and the max player runs the overpowered attack against the classifier chosen by the min player. We call our proposed defense the *Robust Manifold Defense*. If the range of the spanner is a good approximation to the true dataset, a successful defense against adversarial examples in the range of the spanner is sufficient to imply the lack of adversarial examples close to images in the dataset (by the triangle inequality).

Our robust optimization perspective towards adversarial training a la Madry et al. [2017], illustrates the benefit of our proposed method: We are tremendously reducing the dimensionality of the space that the max player is searching over. Rather than searching for adversarial examples inside balls around real images $x \in \mathbb{R}^n$, the max player is searching over pairs of “latent codes” $z, z' \in \mathbb{R}^k$. Lowering the dimensionality of the max player’s search space renders the min-max optimization problem more benign. In particular, targeting this lower-dimensional space makes the max player more effective in identifying images in the range of the spanner on which the classifier’s output differs, and in turn this renders adversarial training more effective. In practice, we may combine normal adversarial training with adversarial training using our overpowered attack, alternating between the two for multiple iterations to reap the benefits of both methods. We use our Robust Manifold Defense to obtain the most robust known MNIST classifier, significantly improving the state of the art against PGD attacks.

² $G(z^*)$ is the projection of an image x to the range of the spanner G iff $z^* = \arg \min_z \|x - G(z)\|$, i.e., $G(z^*)$ is the image in the range of the spanner that is closest to x .

³There are a few key differences in hyperparameters for DefenseGAN used by Athalye et al. [2018] and the original DefenseGAN paper [Samangouei et al., 2018]. We describe the differences in the experimental section.

⁴At the ICML 2018 best paper award talk, N. Carlini poses this as a challenging problem, see Min. 15 of talk. https://nicholas.carlini.com/talks/2018_icml_obfuscatedgradients.mp4

1.1 Contributions

Our contributions are summarized as follows:

- We propose a new type of attack that we call the *overpowered attack*, which searches over pairs of adversarial images in the range of a spanner.
- We show how to use our overpowered attack to *fully* circumvent DefenseGAN [Samangouei et al., 2018]. This resolves an open challenge posed by Athalye et al. [2018].
- We show that our overpowered attack can be combined with adversarial training to increase the adversarial robustness of MNIST classifiers against white box attacks with bounded ℓ_2 norm= 1.5 from a state-of-the-art adversarial accuracy of 91.88% using the TRADES algorithm [Zhang et al., 2019] to an adversarial accuracy of 96.26%. This is a significant increase in a metric that has been proven challenging to improve previously.
- Finally, note that our overpowered attack creates new pairs of adversarial images (that are in the range of the spanner). We emphasize, however, that for evaluating both the circumvention of DefenseGAN and the robustness of our new MNIST classifier, we report adversarial robustness computed on the *original test set images*. Our results are therefore directly comparable to all previous results in the literature.

2 Related work

There is a deluge of recent work on adversarial attacks and defenses. Common defense approaches involve modifying the training dataset such that the classifier is made more robust [Gu and Rigazio, 2014], [Shaham et al., 2015], modifying the network architecture to increase robustness [Cisse et al., 2017] or performing defensive distillation [Papernot et al., 2016b]. The idea of adversarial training [Goodfellow et al., 2014b] and its connection to robust optimization [Shaham et al., 2015, Madry et al., 2017, Sinha et al., 2017] leads to a fruitful line of defenses. On the attacker side, [Carlini and Wagner, 2017a], [Carlini and Wagner, 2017c] show different ways of overcoming many of the existing defense strategies.

Our approach to defending against adversarial examples leverages GANs and VAEs [Goodfellow et al., 2014a, Kingma and Welling, 2013] as spanners. GAT-Trainer by [Lee et al., 2017] uses GANs to perform adversarial training but in a very different way from our work and without projecting on the range of a GAN. MagNet [Meng and Chen, 2017] and APE-GAN [Shen et al., 2017] have the similar idea of denoising adversarial noise using a generative model but use differentiable projection methods that have been successfully attacked by [Carlini and Wagner, 2017b]. Additionally, work by [Song et al., 2018] shows that generative models can be used to construct adversarial examples for classifiers, such that the classifier prediction is very different from a human prediction of the same image.

The most closely related work is DefenseGAN [Samangouei et al., 2018], which we have already compared our work to in length. One of our main contributions is to circumvent it using our overpowered attack. The second related paper is PixelDefend [Song et al., 2017]. This is similar to DefenseGAN except it uses PixelCNN generators. We note that Athalye et al. [2018] has already showed that PixelDefend can be circumvented.

3 An (Ineffective) First Take at Adversarial Defense with Spanners

Given a classifier C_θ parameterized by a vector of parameters θ , an initial idea might be to defend it by finding the projection of the input to be classified to the range of a spanner G , such as a Generator from a GAN or a Decoder from a VAE, and apply the classifier to the projection. More precisely, for some hyper-parameter η and given an input x , we perform the following procedure that we call *Invert and Classify (INC)*:⁵

1. Perform gradient descent in z (latent code) space to minimize $\|G(z) - x\|_2$. Let z^* be the point returned by gradient descent. (Ideally, we would want $z^* \equiv \arg \min_z \|G(z) - x\|_2$, but we settle with whatever gradient descent returns.)
2. If the “projection” $G(z^*)$ of x in the range of the spanner G computed in Step 1 is far from x , i.e. if $\|G(z^*) - x\|_2 \geq \eta$, we reject the input x as “unnatural,” since it lies far from the range of the spanner.
3. Otherwise, we apply our classifier on the projected input, outputting a class label according to $C_\theta(G(z^*))$.

At first glance, this may seem like a reasonable defense, since first-order methods such as PGD or FGSM will be unable to compute the gradient through the projection step due to its non-differentiability. However, as we show in the following section, this is not the case. Indeed, the INC defense belongs to precisely the same class of techniques used by DefenseGAN [Samangouei et al., 2018] and PixelDefend [Song et al., 2017], two defenses which were both found to be circumventable with a simple white-box adversary [Athalye et al., 2018], albeit DefenseGAN had not been fully circumvented prior to our work.

4 The Overpowered Attack

In the previous section, we considered the simplest way of exploiting neural network-based spanners to defend against adversarial attacks. We find, however, that this “projection-based” approach to defend classifiers is ineffective against white-box adversaries. In particular, we propose a new attack which we refer to as “overpowered attack.” We show that the overpowered attack successfully reduces the accuracy of the INC defense from the previous section to 0%. Then, we show that we can slightly modify the overpowered attack to *fully circumvent* the DefenseGAN defense.

Deriving the Attack. Given some perturbation distance ε , one way to attack the (η, G) -INC defense of some classifier C_θ is to find a real image x that is η close to the range of G , as well as another input x' that is both ε -close to x and also within η from the range of the spanner, so that the classifications of the projections $G(z), G(z')$ of x, x' according to C_θ are significantly different.

If such (x, x') exist, however, then (by triangle inequality) there must exist z and z' such that $G(z)$ and $G(z')$ are $(2\eta + \varepsilon)$ -close, yet $C_\theta(G(z))$ and $C_\theta(G(z'))$ are far. The following optimization problem captures the furthest $C_\theta(G(z))$ and $C_\theta(G(z'))$ can be under some loss function \mathcal{L} of interest (e.g. cross-entropy), subject to our derived bound, in terms of ε and η , on the distance between $G(z)$

⁵Note that this is not a defense we propose to use, but a prop to develop our attack in the next section.

and $G(z')$.

$$\sup_{z, z'} \mathcal{L}(C_\theta(G(z)), C_\theta(G(z'))), \quad (1)$$

$$\text{s.t. } \|G(z') - G(z)\|_2^2 \leq (2\eta + \varepsilon)^2. \quad (2)$$

As per our discussion, the above optimization problem upper-bounds how much an attack to INC can change the output. It is an upper bound in a very strong sense. In particular,

- if the value of the above optimization problem is V^* then this means that for *any* real image x , *any* z returned by Step 1 of INC on x , *any* image x' that is ε -close to x , and *any* z' that is returned by Step 1 of INC on x' , the loss from the output of INC on x and x' is at most V^* ;
- this means, in particular, that if V^* is small, then INC suffers loss at most V^* *even accounting for the fact that INC uses gradient descent in Step 1 to project to the range of G , and as such this step may be suboptimal*;
- as such, the above optimization problem captures the worst loss that INC may suffer from an “overpowered adversary,” who can adversarially control also what happens in Step 1 of INC for different inputs.

Summarizing the above points, the above optimization problem serves as an upper bound on the loss from both adversarial attacks and the suboptimality in Step 1 of INC.

Using our overpowered attack, we achieve the following.

Circumventing DefenseGAN. DefenseGAN [Samangouei et al., 2018] is a proposed adversarial defense technique that operates quite similarly to INC. Athalye et al. [2018] show that by using an attack known as Backwards-Pass Differentiable Approximation (BPDA), it is possible to reduce the adversarial accuracy of DefenseGAN from the claimed 96% accuracy to 55% accuracy. This attack was completed under an ℓ_2 threat model with an ℓ_2 per-pixel distortion budget of 0.005.

By combining the overpowered attack that we outlined above with the expectation-over-transformation attack of Athalye et al. [2017], we can successfully reduce the classification accuracy of DefenseGAN to **3%**, under the same perturbation budget and hyperparameter settings⁶. We also evaluate our attack against DefenseGAN using the hyperparameter settings recommended in the original DefenseGAN paper [Samangouei et al., 2018] and the perturbation budget in Athalye et al. [2018] and find that the classification accuracy drops to **5%**. In addition to resolving a major open problem, this attack demonstrates the potential of the latent-space attack as a more powerful attack for adversarial training. Please see Appendix A for details about the attack hyperparameter settings.

We stress the following. As outlined above, the overpowered attack identifies latent codes z, z' whose images $G(z), G(z')$ are classified by C_θ as differently as possible. There is no guarantee that any of $G(z)$ or $G(z')$ is a real image or close to one. To circumvent DefenseGAN, we need instead to identify adversarial images that are ε -close to *real* images. To use the overpowered attack for this purpose, we need to force $G(z)$ and $G(z')$ to be appropriately close real images. We outline how this can be done in Section 6. In particular, our claimed reduction of DefenseGAN accuracy to **3%** is measured in the standard way, i.e. performing perturbations around *real images*, with the exact same perturbation budget under which Athalye et al. [2018] reduce the DefenseGAN accuracy to 55%.

⁶In fact, we use the *average* ℓ_2 perturbation budget reported by Athalye et al. as the *maximum* one in our attack.

5 The Robust Manifold Defense

The attack of the previous section should cause us to reconsider the use of INC and similar “projection-based” approaches for building adversarial defenses. At the same time, the effectiveness of the overpowered attack to resolve the outstanding challenge of circumventing DefenseGAN motivates us to use the overpowered attack within a robust optimization framework a la Madry et al. [2017] to build adversarial defenses. This is what we explore in this section, proposing for this purpose the following min-max formulation. The outer (inf) player of this formulation is setting the parameters of a classifier, while the inner (sup) player is searching for overpowered attacks:

$$\inf_{\theta} \mu \left(\sup_{\substack{z, z': \|G(z) - G(z')\|_2 \leq \\ (2\eta + 2\varepsilon)^2}} \mathcal{L}(C_{\theta}(G(z)), C_{\theta}(G(z'))) \right) + (1 - \mu) \left(\frac{1}{N} \sum_{i=1}^N \mathcal{L}(y^{(i)}, C_{\theta}(x^{(i)})) \right). \quad (3)$$

The mixing weight $\mu \in (0, 1)$ is some hyperparameter that mixes between two objectives. The first measures how much an overpowered attacker can change the output of the classifier. The second measures how well the classifier performs on the training set $\{x^{(i)}, y^{(i)}\}_{i=1}^N$. \mathcal{L} is the loss function that we are interested in minimizing, e.g. cross-entropy loss. Finally, we choose ε to be the allowed perturbation, and we choose η according to the quality of our spanner G ; in particular, we would like to choose η so that real images are within η from the range of the spanner. The quantity $(2\eta + 2\varepsilon)$ appearing in the constraint of the sup player has a similar justification as the justification of $(2\eta + \varepsilon)$ in our description of the overpowered attack in the previous section, namely: the existence of a true adversarial pair, consisting of a *real image* x and a perturbed image x' for which the classifier has high loss $\mathcal{L}(C_{\theta}(x), C_{\theta}(x'))$ implies, by triangle inequality, the existence of a pair of latent codes z and z' such that $\|G(z) - G(z')\|_2 \leq 2\eta + 2\varepsilon$ such that $\mathcal{L}(C_{\theta}(G(z)), C_{\theta}(G(z')))$ is high. Thus, if we want to protect against the existence of the former types of pairs (x, x') , it suffices to protect against the latter types of pairs (z, z') .

Our adversarial defense approach has the following advantages/disadvantages in comparison to previous min-max formulations of adversarial training, such as [Madry et al., 2017], which search for adversarial examples around real images:

1. Notice that the sup player in our formulation searches for pairs z, z' in *latent space*, which is typically much lower-dimensional compared to image space. As such, the sup player in our formulation faces an easier optimization problem than that facing the sup player in the standard min-max formulation of adversarial training, where the sup player is searching in image space. Given the challenging nature of min-max optimization, a decrease in the dimensionality of one of the two players of the optimization problem could provide big gains in our ability to get good solutions.
2. As discussed in length in the previous section, the overpowered attack used by the sup player of our formulation is significantly more powerful than perturbing images from the training set. In effect, it allows the attacker to find adversarial *pairs of images* x, x' , such that neither x nor x' need to be in the training set, and which are close to each other, yet result in classifier outputs that are far. The increased power of the adversary could make a big difference in the robustness of the classifier, since, as also suggested by Madry et al. [2017], endowing the sup player of adversarial training with the ability to perform stronger attacks could yield a

stronger defense. Note also that it is meaningless to search over pairs of inputs x, x' in the ambient space of images, as these are mostly garbage. Searching over pairs of images is only made possible by using the spanner.

3. The disadvantage of our approach is that it needs a good spanner G . The better the spanner, the lower we could choose η and the closer the range of G would be to real images. The worse the spanner, the higher we would need to choose η and the further the range of G might be from real images. In this case, our attacker will be overly powerful, which could make our classifier overly defensive, which might decrease its adversarial robustness with respect to real images.

We use our adversarial training procedure to train robust classifiers for the MNIST [LeCun et al., 1998] and CELEBA [Liu et al., 2015] datasets, and report our findings in Section 6. A main contribution of our approach is that we improve the adversarial robustness of MNIST classifiers against white box attacks with bounded ℓ_2 norm= 1.5 from a state-of-the-art adversarial accuracy of 91.88% using the TRADES algorithm [Zhang et al., 2019] to an adversarial accuracy of 96.26%.

6 Experiments

6.1 Breaking DefenseGAN Additional Details

In this section we show that we can use our latent space attack combined with the expectation-over-transformation attack [Athalye et al., 2017] to break DefenseGAN [Samangouei et al., 2018]. It is worth noting that this attack has some modifications from the straightforward overpowered attack. In the overpowered attack, we search for a pair (z, z') such that $G(z), G(z')$ are close but the classifier makes different predictions on them. However, when attacking DefenseGAN, we do not have the freedom of choosing a pair of images which are classified differently. Instead we are given a *fixed image* x from a dataset and must adversarially perturb it such that DefenseGAN is fooled by the attack.

Our approach to breaking DefenseGAN is as follows: say we are given an image x with true class ℓ , and the GAN G . We wish to find a latent \hat{z} satisfying $\frac{\|G(\hat{z}) - x\|_2}{784} \leq 0.0051$ (this corresponds to a ℓ_2 perturbation of 0.0051 per pixel, as in Athalye et al. [2017]), such that when $G(\hat{z})$ is provided as input to the DefenseGAN mechanism, the classifier has different predictions for x and $G(\hat{z})$.

We will now describe our algorithm for computing \hat{z} . Assuming we are given the GAN G , and the classifier C , we solve the following optimization problem:

$$\hat{z} = \arg \max_{z \in \mathbb{R}^k} \mathbb{E}_{\tau \sim \mathcal{N}(0,1)} \left[\mathcal{L} \left(C \left(G(z) + \frac{\sigma \tau}{\|\tau\|_2} \right), \ell \right) \right], \quad (4)$$

$$\text{s.t.} \quad \frac{\|G(z) - x\|_2}{784} \leq 0.0051, \quad (5)$$

where $\mathcal{L}(C(x'), \ell) := \arg \max_{\ell' \neq \ell} C_{\ell'}(x') - C_{\ell}(x')$, is the Carlini-Wagner loss [Carlini and Wagner, 2017c] and $C_{\ell'}(x'), C_{\ell}(x')$ are the classifier logits for classes ℓ', ℓ on the image x' . The random normal noise $\tau \sim \mathcal{N}(0,1)$ and the corresponding expectation over τ forms the expectation-over-transformation [Athalye et al., 2017] part of our attack. It is necessary to make our attack robust to noise implicit in the DefenseGAN procedure- if we provide as input an image $G(\hat{z})$, DefenseGAN may recover a latent code which is close to, but not exactly \hat{z} . To make sure that the code recovered

Algorithm 1 Pseudo-code for Defense-GAN break

Input: Image x , label ℓ , classifier C , GAN G

Hyperparameters: Step-size η , number of restarts R , number of gradient steps T , noise variance σ .

Find inverse of x , i.e., $z^* = \arg \min_z \|x - G(z)\|_2$.

$\mathcal{L}(C(x'), \ell) := \arg \max_{\ell' \neq \ell} C_{\ell'}(x') - C_{\ell}(x')$, where $C_{\ell'}(x')$ is the logit for class ℓ' on the image x' .

// Do R restarts and pick the best restart.

for $r \leftarrow 1$ to R **do**

Set $z_r(0) \leftarrow z^*$.

Set $\lambda_r(0) \leftarrow 10^3$.

// Do T steps of gradient descent-ascent.

for $t \leftarrow 0$ to $T - 1$ **do**

Sample $\tau_j \sim \mathcal{N}(0, 1), j = 1, \dots, 100$.

$\text{loss}_r(t) \leftarrow \lambda_r(t) \cdot \left(0.0051 - \frac{\|G(z_r(t)) - x\|}{784}\right) + \sum_{j=1}^{100} \mathcal{L}\left(C\left(G(z_r(t)) + \sigma \frac{\tau_j}{\|\tau\|}\right), \ell\right)$.

$z_r(t+1) \leftarrow z_r(t) + \eta \nabla_z \text{loss}_r(t)$. // ascent step in z .

$\lambda_r(t+1) \leftarrow \lambda_r(t) - \eta \nabla_{\lambda} \text{loss}_r(t)$. // descent step in λ .

end for

end for

$r^* \leftarrow \arg \max_r \mathbb{E}_{\tau \sim \mathcal{N}(0,1)} \left[\mathcal{L}\left(C\left(G(z_r(T)) + \sigma \frac{\tau}{\|\tau\|}\right), \ell\right) \right]$ // restart that had largest CW-loss.

Output: $G(z_{r^*}(T))$.

by DefenseGAN produces a misclassified image, we introduce noise of our own and average over it, which ensures that it is robust to noise in optimization introduced by DefenseGAN. To this end, we introduce the scaling constant σ in (4) which can be tuned to account for noise introduced by the DefenseGAN procedure.

By introducing a Lagrange multiplier λ , we can rewrite the above problem as an equivalent max-min optimization problem:

$$\max_{z \in \mathbb{R}^k} \min_{\lambda \geq 0} \mathbb{E}_{\tau} \left[\mathcal{L}\left(C\left(G(z) + \frac{\sigma \tau}{\|\tau\|_2}\right), \ell\right) \right] + \lambda \left(0.0051 - \frac{\|G(z) - x\|}{784}\right). \quad (6)$$

We find that this max-min optimization problem is tractable by doing gradient descent ascent in the primal variable z and Lagrange multiplier λ . See Algorithm 1 for pseudo-code detailing our attack and section A in the Appendix for hyperparameters used in optimization.

Empirical Results: When we evaluate our latent space attack (4) with $\sigma = 0.5$ on the MNIST *test set*, under the same hyperparameter settings for DefenseGAN used by Athalye et al. [2018], we find that DefenseGAN is robust to only **3%** of our attacks. We emphasize that if x_i is the i^{th} image in the MNIST test set, then our attack $G(\hat{z}_i)$ satisfies the perturbation constraint in the threat model. Of the 10000 images in the MNIST test set, we can find attacks such that 97% of them fool DefenseGAN. Figure 1 shows some examples of original images and perturbed images which break DefenseGAN.

Samangouei et al. [2018] and Athalye et al. [2018] have a few differences in hyperparameters employed within DefenseGAN- the number of gradient descent steps and number of restarts used to

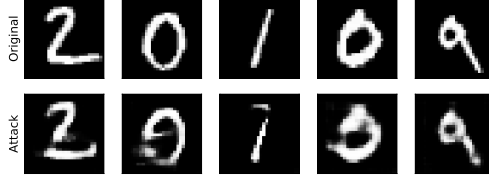


Figure 1: Attacks against DefenseGAN. The top row shows images from the MNIST test set and the bottom row shows their corresponding perturbed versions. The attacks constructed in the bottom row satisfy the perturbation constraint imposed by Athalye et al. [2018] and also successfully break DefenseGAN, i.e. are misclassified.

invert an image are two key parameters in DefenseGAN. Samangouei et al. [2018] use 200 gradient steps with 10 random restarts, whereas Athalye et al. [2018] use 20,000 steps. As noted in [Samangouei et al., 2018], increasing the number of gradient descent steps can decrease classification accuracy, since adversarial noise may be reconstructed. The use of only 200 gradient steps in DefenseGAN implies that the projection step has a lot of variance in the inversion step, i.e., the reconstructions obtained over multiple runs of DefenseGAN can vary a lot. To make our attack robust to this variance, we run a loop over different values of $\sigma = [1, 1.5, 2, 2.5, 3, 3.5]$ in (4) and find that the classification accuracy of DefenseGAN drops to 5%. Please see Appendix A for more details about the hyperparameters we used for DefenseGAN.

6.2 Adversarial Training using the Overpowered Attack

It is well known that a strong attack against a classifier can be used to improve its robustness through adversarial training [Shaham et al., 2015, Madry et al., 2017]. Inspired by this, we verify that our proposed overpowered attack can be used to boost the robust accuracy obtained by Madry et al. on the MNIST dataset.

We first run the adversarial training procedure proposed by Madry et al. [2017] to get a base classifier $C^{(0)}$. The threat model allows white box access and perturbations whose ℓ_2 norm is at most $\delta = 1.5$. During adversarial training, the adversary uses 40 step PGD with step size 0.075 and 1 random perturbation. Additionally, all pixel values are in the range $[0, 1]$.

Starting with a robust classifier $C_{\theta}^{(0)}$ trained using the algorithm by Madry et al. [2017], we employ our overpowered attack to improve its robustness in the following way:

1. use the overpowered attack to find a batch of 50 pairs $\{(z_i, z'_i)\}_{i=1}^{50}$. We then modify the parameters of the network, θ , by performing a single gradient descent step on the function $10^{-2} \cdot \left(\frac{1}{50} \sum_{i=1}^{50} \mathcal{L}(C_{\theta}^{(0)}(G(z_i)), C_{\theta}^{(0)}(G(z'_i))) \right)$, where \mathcal{L} is the cross-entropy loss.
2. Rerun the adversarial training algorithm in Madry et al. [2017] for 5 epochs, starting with the parameters obtained after Step 1.
3. Repeat step 1 using $C_{\theta}^{(1)}$.

We observe that running the above loop approximately 10 times returns a classifier whose robustness against white box access, bounded ℓ_2 norm perturbations of the MNIST test set is 8% more than the initial robust model $C^{(0)}$. We would also like to emphasize that the spanner model



Figure 2: Pairs of images $G(z)$ and $G(z')$ generated by our overpowered attack, such that they are classified differently by a model trained using the adversarial training algorithm in Madry et al. [2017]. These images are used for adversarial training by penalizing the cross entropy loss of the classifier’s prediction on pairs of images. We observe that using these images in conjunction with samples from Madry et al. [2017] improves robustness. Note that the final evaluation of robustness is performed on the actual MNIST test set images.

Attack	Madry et al.	TRADES	Ours
PGD (40 steps, $\delta = 1.5$)	89.93%	91.88%	96.26%
PGD (100 steps, $\delta = 1.5$)	89.87%	91.82%	96.25%
PGD (40 steps, $\delta = 2.5$)	76.09%	69.59%	95.21%
PGD (100 steps, $\delta = 2.5$)	75.90%	68.86%	95.16%

Table 1: Robustness of different models against PGD adversaries with white box access for perturbations with ℓ_2 norm 1.5, 2.5 on the MNIST test set. Our algorithm improves upon the current state of the art by at least 5%. Although it was trained against perturbations whose ℓ_2 norm was at most $\delta = 1.5$, we notice that the robustness does not fall significantly even when we increase the budget to $\delta = 2.5$.

was trained only on images in the MNIST training set, and contains no information about the MNIST test set. Please see Section B in the Appendix for more details about hyperparameters, train/validation set, etc.

The intuition for why the overpowered attack helps robustness is as follows: the algorithm by Madry et al. [2017] finds adversarial perturbations on images in the training set. If it is run long enough, we observe that the model is capable of achieving 100% robustness against the adversary on the training set, but the robustness on the test set plateaus at 90% after roughly 10 – 20 epochs. If we run the overpowered attack on this classifier, it will produce images as shown in Figure 2 that trick the classifier and are not based on images in the training set. Our hypothesis is that these samples allow us to meaningfully perturb the classifier, such that if we rerun the algorithm employed by Madry et al. starting from the perturbed classifier, we reach a new local minimum with improved robustness.

Results: We report our empirical results in Table 1. We compare our model’s robustness against models trained using the TRADES algorithm [Zhang et al., 2019] with hyperparameter $\beta = 6$ (we did a hyperparameter search over $\beta = 1, 6, 10$) and the PGD training in [Madry et al., 2017]. All models were trained against adversaries that had white box access and perturbations had bounded ℓ_2 norm = 1.5. The reported numbers are for bounded ℓ_2 perturbations of images in the MNIST test set. It is interesting to see that although our model was trained using a threat model where the adversarial perturbation budget was $\delta = 1.5$, it has good robustness even when the perturbation



Figure 3: Pairs of images $G(z)$ and $G(z')$ generated by our overpowered attack on a non robust gender classifier on the CelebA dataset. These images are very close but the confidence of the classifier changes drastically. These pairs of images are adversarial attacks for this classifier which lie on the manifold of a generator.

budget for the adversary on the test set is increased to $\delta = 2.5$.

6.3 Adversarial Training on the CelebA Dataset

In this section we show that we can use the adversarial training procedure described in Section 5 to train a robust gender classifier on the CelebA dataset. Figure 3 shows randomly selected successful results of the overpowered attack against a non robust classifier protected by the INC algorithm.

The attacks found with this optimization tend to yield images with semantically relevant features from both classes, and furthermore often introduce meaningful (though minute) differences between $G(z)$ and $G(z')$ (e.g. facial hair, eyes widening, etc.). Also, as confirmed by [Athalye et al., 2018], none of these images actually induce different classifications on the end-to-end classifier, which we attribute to imperfections in the projection step of the defense (that is, since $G(z^*) \neq x$ exactly). However, we consider this implementation, and we robustify the classifier against this attack. We use a variant of the more complex min-max optimization proposed in Section 5.

After 10,000 iterations, **100%** of the images produced by the overpowered attack were valid, but with **22%** of them inducing different classification, and an average KL divergence of **0.08**, showing that the classifier has softened its decision boundary.

We also feed the inputs generated by the overpowered attack on the initial classifier into the adversarially trained classifier. Figure 4 shows a randomly selected subset of these examples with their respective classifier output. Please see Section C in the Appendix for a more detailed description of the adversarial training procedure.

7 Conclusion

We show how we can use generative models to create a new overpowered attack that searches over pairs of images in the spanner range. Our attack improves the state of the art for DefenseGAN resolving a challenging problem in the field.

Further, we show that the generated attack images can be used to boost robustness of existing adversarial training procedures and can also be used to train robust classifiers that display natural uncertainty around decision boundaries. For MNIST we increase the ℓ_2 robustness compared to the best previously known and for CelebA we show that there can be adversarial examples on the natural image spanner range. The main limitation of our approach is that it relies on a good spanner or generative model for the domain of interest.

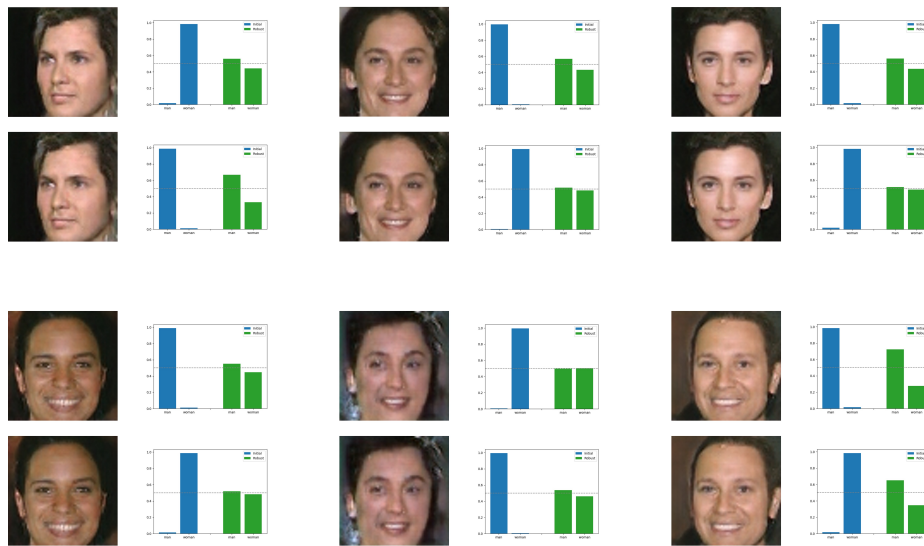


Figure 4: The softmax output of both the original (blue) and robust adversarially trained (green) classifier on the images generated by the attack on the non-robustified classifier. As shown, the robust projection defense makes the classifier reduce its confidence on such borderline images.

References

- Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397*, 2017.
- Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.
- David Berthelot, Tom Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.
- Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. *arXiv preprint arXiv:1705.07263*, 2017a.
- Nicholas Carlini and David Wagner. MagNet and “efficient defenses against adversarial attacks” are not robust to adversarial examples. *arXiv preprint arXiv:1711.08478*, 2017b.
- Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 39–57. IEEE, 2017c.
- Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, pages 854–863, 2017.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014a.
- Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014b.
- Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014.
- Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- Hyeungill Lee, Sungyeob Han, and Jungwoo Lee. Generative adversarial trainer: Defense to adversarial perturbations with gan. *arXiv preprint arXiv:1705.03387*, 2017.
- Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

- Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.
- Dongyu Meng and Hao Chen. MagNet: a two-pronged defense against adversarial examples. *arXiv preprint arXiv:1705.09064*, 2017.
- Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against deep learning systems using adversarial examples. *arXiv preprint arXiv:1602.02697*, 2016a.
- Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *Security and Privacy (SP), 2016 IEEE Symposium on*, pages 582–597. IEEE, 2016b.
- Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. *International Conference on Learning Representations*, 2018. URL <https://openreview.net/forum?id=BkJ3ibb0->.
- Uri Shaham, Yutaro Yamada, and Sahand Negahban. Understanding adversarial training: Increasing local stability of neural nets through robust optimization. *arXiv preprint arXiv:1511.05432*, 2015.
- S. Shen, G. Jin, K. Gao, and Y. Zhang. Ape-gan: Adversarial perturbation elimination with gan. *ICLR Submission, available on OpenReview*, 2017.
- Aman Sinha, Hongseok Namkoong, and John Duchi. Certifiable distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017.
- Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766*, 2017.
- Yang Song, Rui Shu, Nate Kushman, and Stefano Ermon. Constructing unrestricted adversarial examples with generative models. In *Advances in Neural Information Processing Systems*, pages 8312–8323, 2018.
- Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.
- Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. *arXiv preprint arXiv:1901.08573*, 2019.
- Jian Zhang, Ioannis Mitliagkas, and Christopher Ré. Yellowfin and the art of momentum tuning. *arXiv preprint arXiv:1706.03471*, 2017.

A Hyperparameters for DefenseGAN break

The generator and classifier architecture/checkpoint as well as an implementation of the attack can be found in the GitHub repository at <https://github.com/ajiljalal/manifold-defense>.

For the initial value of the variable z in Eqn (6), we first find the latent z^* which produces an image in the range of G which is closest to x , i.e., $z^* := \arg \min_{z \in \mathbb{R}^k} \|x - G(z)\|$. We initialize λ with the value -1000 . To solve the optimization in Eqn 6, we use an Adam optimizer [Kingma and Ba, 2014] with learning rate 0.05 for z and an SGD optimizer with learning rate 10000 for λ .

A.1 Hyperparameters for DefenseGAN

We evaluate DefenseGAN under two hyperparameter settings:

- Athalye et al. [2018]: In this setting, the projection operation in DefenseGAN uses 20,000 gradient steps with 20 random restarts. The reconstruction closest in ℓ_2 to the original image is fed to the classifier as input. The projection step uses an SGD optimizer with learning rate=500 and momentum=0.7. Under this threat model, our combined attack (overpowered attack + expectation-over-transformation) reduces the accuracy of DefenseGAN to **3%**.
- Samangouei et al. [2018]: In this setting, the projection operation in DefenseGAN uses 200 gradient steps with 10 random restarts. The reconstruction closest in ℓ_2 to the original image is fed to the classifier as input. The projection step uses an SGD optimizer with learning rate=10 and momentum=0.7. Under this threat model, our combined attack of overpowered attack and expectation-over-transformation (EOT) reduces the accuracy of DefenseGAN to **5%**. In this setting, we also grid-search over the EOT robustness parameter *for each image*.

B Hyperparameters for Boosting Madry’s MNIST

For the max-min optimization, we use the Yellowfin optimizer Zhang et al. [2017] for z and λ with learning rate 10^{-4} . We perform 500 updates for the overpowered attack. The classifier is trained with an adam optimizer with learning rate 10^{-4} .

We use the first 55,000 images in the MNIST dataset for training the VAE and classifier. We use samples 55,000-60,000 as validation data. We use validation data to pick the classifier with maximum robust accuracy. The test set was the standard MNIST test set.

The MNIST classifier was the same used by Madry et al. Madry et al. [2017], and we ran their code for 200 epochs against a PGD adversary, and the perturbations had bounded ℓ_2 norm= 1.5. The generative model was a VAE [Kingma and Welling, 2013] from <https://github.com/pytorch/examples/blob/master/vae/main.py> except the decoder had architecture had 500 nodes in the first layer, 500 nodes in the second layer and the final output had width 784. We used a LeakyReLU as the non-linearity in the hidden layers.

The model definitions and checkpoints for the classifier and generator can be found at <https://github.com/ajiljalal/manifold-defense>.

The frequency with which we perform the overpowered attack is crucial to stabilize the adversarial training procedure. Using too many samples from the overpowered attack can cause the classifier to overfit to artifacts produced by the generator, and hence it may end up random guessing on the train/test set. We experimented with how many epochs of PGD training we should do before we

perform the overpowered attack once. We tried running 3; 5; 8 epochs of regular PGD training in Madry et al. [2017] and then performing the overpowered attack to generate adversarial pairs. We found that running 5 epochs of PGD followed by one batch of overpowered attack samples provided best results. Selection of the best model was done by evaluating robustness on the validation set.

C Appendix for Adversarial Training on CelebA

In this section we show that we can use the adversarial training procedure described in Section 5 to train a robust gender classifier on the CelebA dataset. As described in section 4, we perform an overpowered attack on the standard invert-and-classify architecture. We search for z and z' such that $G(z)$ and $G(z')$ are close but induce dramatically different classification labels. Recall that this involves solving a max-min optimization problem:

$$\sup_{z, z'} \inf_{\lambda \leq 0} \|C_\theta(G(z)) - C_\theta(G(z'))\|_2^2 + \lambda \cdot (\|G(z) - G(z')\|_2^2 - (2\eta + \varepsilon)^2).$$

In practice, we set our ℓ_2 constraint to $(2\eta + \varepsilon)^2 \approx 2.46$, corresponding to an average squared difference of $2 \cdot 10^{-4}$ per pixel-channel. We implement the optimization through alternating iterated gradient descent on both λ and (z, z') , with a much more aggressive step size for the λ -player (since its payoff is linear in λ). The gradient descent procedure is run for 10,000 iterations. Because the ℓ_2 constraint was imposed through a Lagrangian, we consider two z, z' valid if the mean distance between the images is $< 5 \cdot 10^{-4}$. The optimization terminated with **93%** of the images satisfying the ℓ_2 constraint; within this set, the average KL-divergence between classifier outputs was **2.47**, with **57%** inducing different classifications. Figure 3 shows randomly selected successful results of the attack.

The attacks found with this optimization tend to yield images with semantically relevant features from both classes, and furthermore often introduce meaningful (though minute) differences between $G(z)$ and $G(z')$ (e.g. facial hair, eyes widening, etc.). Also, as confirmed by [Athalye et al., 2018], none of these images actually induce different classifications on the end-to-end classifier, which we attribute to imperfections in the projection step of the defense (that is, since $G(z^*) \neq x$ exactly). However, we consider this implementation, and we robustify the classifier against this attack. We use a variant of the more complex min-max optimization proposed in Section 5:

$$\inf_{\theta} \mu \left(\sup_{\substack{z, z': \|G(z) - G(z')\|_2^2 \leq \\ (2\eta + 2\varepsilon)^2}} \|C_\theta(G(z)) - C_\theta(G(z'))\|_2^2 \right) + (1 - \mu) \left(\frac{1}{N} \sum_{i=1}^N \mathcal{L}(y^{(i)}, C_\theta(x^{(i)})) \right), \quad (7)$$

where \mathcal{L} is the cross-entropy loss.

We implement this through *adversarial training* [Madry et al., 2017]; at each iteration, in addition to sampling a cross-entropy loss from images from the dataset, we also sample an adversariality loss, where we generate a batch of “adversarial” inputs using 500 steps of the min-max attack, then add the final ℓ_2 distance between the classification outputs to the cross-entropy loss. Please see Section C in the Appendix for more details about the adversarial training procedure. As shown in Figure 6, the classifier learns to minimize the adversary’s ability to find examples. After robustifying the classifier using this adversarial training, we once again try the attack described earlier in this section for the same 10,000 iterations. Figure 6 in the Appendix shows the convergence of the attack

against both the initial and adversarially trained classifier for two values of η^2 , showing the inefficacy of the attack on the adversarially trained classifier. After 10,000 iterations, **100%** of the images were valid, but with **22%** of them inducing different classification, and an average KL divergence of **0.08**, showing that the classifier has softened its decision boundary.

In Table 2 in we see that the robust classifier is effective against the overpowered latent space attack, which is an attack that is crafted for the *INC protected classifier*.

The adversarial training does not significantly impact classification accuracy over the standard classifier: on normal input data, the model achieves the same **97%** accuracy undefended. We also feed the inputs generated by the min-max attack on the initial classifier into the adversarially trained classifier, and observe that the average classification divergence between examples drops to **0.007**, with only **18%** of the valid images being classified inconsistently.

Figure 4 shows a randomly selected subset of these examples with their respective classifier output.

We implement this through *adversarial training* [Madry et al., 2017]; at each iteration, in addition to sampling a cross-entropy loss from images from the dataset, we also sample an adversariality loss, where we generate a batch of “adversarial” inputs using 500 steps of the min-max attack, then add the final ℓ_2 distance between the classification outputs to the cross-entropy loss. As shown in Figure 6 in the Appendix, the classifier learns to minimize the adversary’s ability to find examples. After robustifying the classifier using this adversarial training, we once again try the attack described earlier in this section for the same 10,000 iterations. Figure 6 in the Appendix shows the convergence of the attack against both the initial and adversarially trained classifier for two values of η^2 , showing the inefficacy of the attack on the adversarially trained classifier. After 10,000 iterations, **100%** of the images were valid, but with **22%** of them inducing different classification, and an average KL divergence of **0.08**, showing that the classifier has softened its decision boundary.

In Table 2 we see that the robust classifier is effective against the overpowered latent space attack, which is an attack that is crafted for the *INC protected classifier*.

The adversarial training does not significantly impact classification accuracy over the standard classifier: on normal input data, the model achieves the same **97%** accuracy undefended. We also feed the inputs generated by the min-max attack on the initial classifier into the adversarially trained classifier, and observe that the average classification divergence between examples drops to **0.007**, with only **18%** of the valid images being classified inconsistently.

C.1 Architecture

Generator: We use a BEGAN Berthelot et al. [2017] and the Tensorflow repository from <https://github.com/carpedm20/BEGAN-tensorflow> as the generator.

Classifier We modified the CIFAR10 classifier from https://www.tensorflow.org/tutorials/images/deep_cnn to work on the CelebA dataset Liu et al. [2015] such that the final layer has 2 nodes for binary classification of gender.

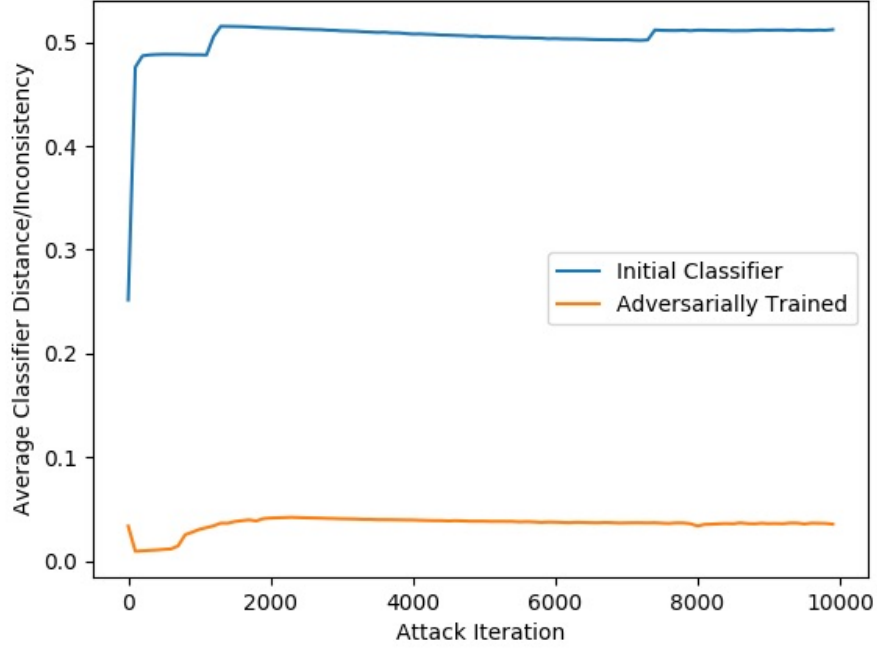


Figure 5: The average $\|C(G(z)) - C(G(z'))\|_2$ for pairs (z, z') found by the attack.

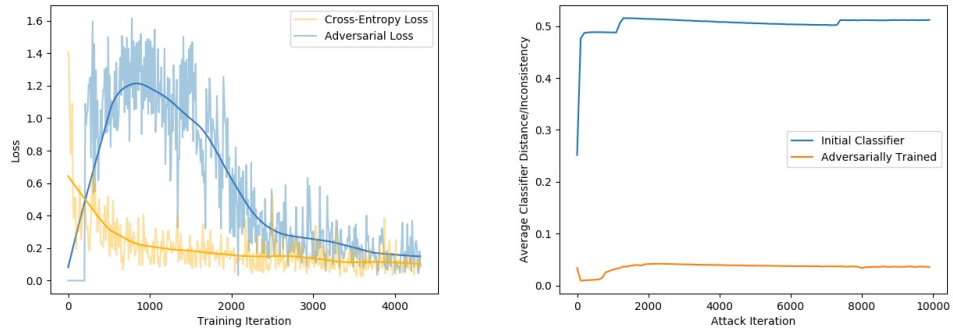


Figure 6: The cross-entropy and adversarial components of the loss decaying as training continues.

Table 2: Accuracy of the celebA classifier under different attacks: our overpowered attack, FGSM [Goodfellow et al., 2014b], BIM [Kurakin et al., 2016], PGD [Madry et al., 2017] at various powers ϵ . **NR** refers to **non robust classifier**, **NR+INC** refers to **non robust classifier with INC** and **R+INC** refers to **robust classifier with INC**.

Attack	CelebA		
	NR	NR+INC	R+INC
Clean Data	97%	84%	90%
Overpowered attack	0%	0%	90%
FGSM ($\epsilon = 0.05$)	1%	82%	86%
FGSM ($\epsilon = 0.1$)	0%	80%	77%
BIM ($\epsilon = 0.05$)	0%	71%	85%
BIM ($\epsilon = 0.1$)	0%	63%	76 %
PGD ($\epsilon = 0.05$)	0%	70%	84%
PGD ($\epsilon = 0.1$)	0%	62%	75 %