DISENTANGLING ADVERSARIAL ROBUSTNESS IN DIRECTIONS OF THE DATA MANIFOLD

Anonymous authors
Paper under double-blind review

ABSTRACT

Using generative models (GAN or VAE) to craft adversarial examples, i.e. generative adversarial examples, have received attention in recent years. Previous studies showed experimentally that the generative adversarial examples work differently compared to that of the regular adversarial examples in many aspects, such as attack rates, perceptibility, and generalization. However, few works focus on the theoretical analysis of the attacking mechanisms of the two kinds of adversarial examples and the reasons for the existence and pervasiveness of regular and generative adversarial examples are unclear. In this work, we provide a theoretical study on this problem and show that adversarial robustness can be disentangled in directions of the data manifold. Specifically, we claim that: 1, Regular adversarial examples leave the data manifold towards the small variance directions. 2, Standard adversarial training increases model robustness by extending the boundary in the small variance directions. 3, Generative adversarial examples attack towards the large variance directions. 4, Adversarial training with generative adversarial examples increases model robustness by extending the boundary in the large variance directions. To confirm our findings, theoretically, we study the Gaussian mixtures model case. We give the claims based on the excess risk and optimal saddle point analysis of the minimax problem of adversarial training with generative models. Experimentally, we show that these phenomena also exist in real datasets.

1 INTRODUCTION

In recent years, deep neural networks (DNNs) (Krizhevsky et al. (2012); Hochreiter and Schmidhuber (1997)) have become popular and successful in many machine learning tasks. They have been used in different problems with great success. But DNNs are shown to be vulnerable to adversarial examples (Szegedy et al. (2013); Goodfellow et al. (2014a)). A well-trained model can be easily attacked by adding a small perturbation to the image. An effective way to solve this issue is to train the robust model using training data augmented with adversarial examples, i.e. adversarial training.

With the growing success of generative models, researchers have tried to use Generative adversarial networks (GAN) (Goodfellow et al. (2014b)) and Variational autoencoder (VAE) (Kingma and Welling (2013)) to generate adversarial examples (Xiao et al. (2018); Zhao et al. (2017); Song et al. (2018a); Kos et al. (2018); Song et al. (2018b)) to fool the classification model with great success. They found that standard adversarial training cannot defend these new attacks. Unlike the regular adversarial examples, these new adversarial examples are perceptible by humans but they preserve the semantic information of the original data. A good DNN should be robust to such semantic attacks. Since the GAN and VAE are approximations of the true data distribution, these adversarial examples will stay in the data manifold. Hence they are called **on-manifold** adversarial examples by (Stutz et al. (2019)). On the other hand, experimental evidence support that regular adversarial examples leave the data manifold (Song et al. (2017)). We call the regular adversarial examples as **off-manifold** adversarial examples.

The concepts of on-manifold and off-manifold adversarial examples are important. Because they can help us to understand the issue of conflict between adversarial robustness and generalization (Stutz et al. (2019); Raghunathan et al. (2019)), which is still an open problem. But the current studies about on-manifold and off-manifold adversarial examples focus on experiments. In our work, we aim to provide a theoretical study of this problem. We consider a generative attack method that adding a

small perturbation in the latent space of the generative models. Since standard adversarial training cannot defend this attack, we consider the training methods that using training data augmented with these on-manifold adversarial examples, which we call latent space adversarial training. Then we compare it to standard adversarial training (training with off-manifold adversarial examples).

Contributions: We study the theoretical properties of latent space adversarial training and standard adversarial training in Gaussian mixtures model, and find that:

- Regular adversarial examples attack towards the small variance direction.
- Standard adversarial training increases the model robustness by amplifying the small variance, hence it extends the boundary in the small variance directions of the data manifold.
- Generative adversarial examples attack towards the large variance directions.
- Latent space adversarial training increases the model robustness by amplifying the large variance, hence it extends the boundary in the large variance directions of the data manifold.

We provide experiments on MNIST and CIFAR10 and show that the about phenomena also exist in real datasets. Adversarial robustness can be disentangled in small variance directions and large variance directions. It gives us a new perspective to understand the behavior of on-manifold and off-manifold adversarial examples.

2 Related work

Our work is related to attacks and defense methods. Specifically, we care about attacks and defense with generative models.

Attack Adversarial examples for deep neural networks were first intruduced in (Szegedy et al. (2013)). However, adversarial machine learning or robust machine learning has been studied for a long time (Biggio and Roli (2018)). In the setting of **white box attack** (Kurakin et al. (2016); Papernot et al. (2016); Moosavi-Dezfooli et al. (2016); Carlini and Wagner (2017)), the attacker have fully access to the model (weights, gradients, etc.). In **black box attack** (Chen et al. (2017); Su et al. (2019); Ilyas et al. (2018)), the attacker have limited access to the model. First order optimization methods, such as PGD (Madry et al. (2017)), are considered state-of-the-art white box attack. Zeroth order optimization methods are used in black box setting.

Generative adversarial examples Recently, generative models have been used to craft adversarial examples (Xiao et al. (2018); Song et al. (2018b); Kos et al. (2018); Schott et al. (2018)). The adversarial examples are more natural (Zhao et al. (2017)). These adversarial examples lie in the data manifold, and they are called **on-manifold** adversarial examples by (Stutz et al. (2019)). Correspondingly, the regular adversarial examples are called **off-manifold** adversarial examples.

Defense Training algorithms against adversarial attacks can be subdivided into the following categories. **Adversarial training:** The training data is argumented with adversarial examples to make the models more robust (Madry et al. (2017); Szegedy et al. (2013); Tramèr et al. (2017)). **Preprocessing:** Inputs or hidden layers are quantized, projected onto different sets or other preprocessing methods (Buckman et al. (2018); (Guo et al. (2017); Kabilan et al. (2018)). **Stochasticity:** Inputs or hidden activations are randomized (Prakash et al. (2018); Dhillon et al. (2018); Xie et al. (2017)). However, many of them have been broken by carefully designed attack (Athalye et al. (2018)). Only adversarial training (Madry et al. (2017)) is effective.

Defense with generative model Using generative models to design defense algorithms have been studied extensively. Using GAN, we can project the adversarial examples back to the data manifold (Jalal et al. (2017); Samangouei et al. (2018)). VAE is used to train robust model (Schott et al. (2018)).

3 Problem description

Original space adversarial training: Consider the classification problem of training a classifer f_θ to map the data points $x \in \mathcal{X} \subset \mathbb{R}^d$ to the labels $y \in \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are the input data space and the label space. The classifier f_θ is parameterized by θ. We assume that the data pairs (x, y)

are sampled from the distribution $P(X, Y)$ over $\mathcal{X} \times \mathcal{Y}$. Standard training is to find the solution of $\min_\theta \mathbb{E}_{(x,y)\sim P}\ell(f_\theta(x), y)$, where $\ell(\cdot,\cdot)$ is the loss function. Adversarial training is to solve the minimax problem

$$\min_\theta \mathbb{E}_{(x,y)\sim P} \max_{\|x-x'\|\leq\varepsilon} \ell(f_\theta(x'), y), \tag{1}$$

where ε is the threshold of perturbation. Here we can use $\ell_1,\ \ell_2$ or ℓ_∞-norm Madry et al. (2017). The inner maximization problem is to find the adversarial examples x' to attack the given classifier f_θ. The outer minimization problem is to train the classifier to defend the given adversarial examples x'. we refer to these attacks as regular attacks. We refer to these minimax problems as standard adversarial training or original space adversarial training.

Latent space adversarial training: We assume that the data lie in a low dimensional manifold of $\mathbb{R}^d$. Furthermore, we assume the true distribution $\mathcal{D}$ is a pushforward from a prior distribution $z \sim \mathcal{N}(0, I)$ using $G(z)$, where $G : \mathcal{Z} \to \mathcal{X}$ is a mapping from the latent space $\mathcal{Z}$ to the original space $\mathcal{X}$. This is a basic assumption of GAN or VAE. Let $I : \mathcal{X} \to \mathcal{Z}$ be the inverse mapping of $G(z)$. Latent space adversarial training is to solve the following minimax problem

$$\min_\theta \mathbb{E}_{(x,y)\sim P} \max_{\|z'-I(x)\|\leq\varepsilon} \ell(f_\theta(G(z')), y). \tag{2}$$

Unlike the regular attacks, the distance between the original examples and adversarial examples can be large. To preserve the label of the data, we use the conditional generative models (e.g. C-GAN (Mirza and Osindero (2014)) and C-VAE (Sohn et al. (2015))), i.e. the generator $G_y(z)$ and inverse mapping $I_y(x)$ are conditioned on the label y, for adversarial training. We refer to these attacks as generative attack. We refer to these adversarial training as latent space adversarial training.

Regular attacks algorithms Two widely used gradient-based attack algorithms for the inner maximization problem in euqation (1) are **fast gradient sign method (FGSM)** (Goodfellow et al. (2014a)) and **projected gradient descend (PGD)** (Madry et al. (2017)). Using FGSM, the adversarial examples are calculated by

$$x' = x + \varepsilon \text{sgn}(\nabla_x \ell(f_\theta(x), y)),$$

where ∇_x denotes the gradient with respect to x. PGD attempts to find a near optimal adversarial example for the inner maximization problem (1) in multiple steps. In the tth step,

$$x^{t+1} = \Pi_{x+S}[x^t + \alpha \text{sgn}(\nabla_x \ell(f_\theta(x^t), y))],$$

where α is the step size, $\Pi_{x+S}[\cdot]$ is the projection operator to project the given vector to the constraint $x + S = \{x' | \|x - x'\| \leq \varepsilon\}$. In the whole paper, we refer to these as FGSM-attack and PGD-attack. We refer to the corresponding original space adversarial training as FGSM-adv and PGD-adv.

Generative attacks algorithm In our experiments, we use FGSM in the latent space for the inner maximization problem in equation (2)

$$z' = I(x) + \varepsilon \text{sgn}(\nabla_z \ell(f_\theta(G(z)), y)).$$

Because of the mode collapse issue of GAN, adding a small perturbation in the latent space of GAN may output the same images. We use VAE in our experiments. we refer to this generative attack and latent space adversarial training as VAE-attack and VAE-adv.

4 Theoretical analysis

In this section, we study the difference between adversarial training in the latent space and in the original space. We study the simple binary classification setting proposed by Ilyas et al. (2019). The main reason for using this model is that we can find the optimal closed-form solution. For a more complex model, we can only solve it numerically. The closed-form solution gives us insights to understand adversarial training.

4.1 Theoretical model setup

Gaussian mixtures (GM) model Assume that data points (x, y) are sampled according to $y \sim \{-1, 1\}$ unifomly and $x \sim \mathcal{N}(y\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$, where $\boldsymbol{\mu}_*$ and $\boldsymbol{\Sigma}_*$ denote the true mean and covariance matrix of the data distribution. For the data in the class $y = -1$, we replace x by $-x$, then we can view the whole dataset are sampled from $\mathcal{D} = \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$.

Classifier The goal of standard training is to learn the parameters $\boldsymbol{\Theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ such that

$$\boldsymbol{\Theta} = \arg\min_{\boldsymbol{\mu},\boldsymbol{\Sigma}} \mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \arg\min_{\boldsymbol{\mu},\boldsymbol{\Sigma}} \mathbb{E}_{x\sim\mathcal{D}}[\ell(x; \boldsymbol{\mu}, \boldsymbol{\Sigma})], \tag{3}$$

where $\ell(\cdot)$ represents the negative log-likelihood function. Adversarial training is to find

$$\boldsymbol{\Theta_r} = \arg\min_{\boldsymbol{\mu},\boldsymbol{\Sigma}} \mathcal{L}_r(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \arg\min_{\boldsymbol{\mu},\boldsymbol{\Sigma}} \mathbb{E}_{x\sim\mathcal{D}}[\max_{\|x-x'\|\leq\varepsilon} \ell(x'; \boldsymbol{\mu}, \boldsymbol{\Sigma})]. \tag{4}$$

We use $\mathcal{L}$ and $\mathcal{L}_r$ to denote the standard loss and adversarial loss. After training, we classify a new data point x to the class $\text{sgn}(\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}x)$.

Generative model In our theoretical study, We use a linear generative model, probabilistic principle components analysis (P-PCA) (Tipping and Bishop (1999)). P-PCA can be viewed as linear GAN (Feizi et al. (2017); Feizi et al. (2020)) and linear VAE (Dai et al. (2017)).

Given dataset $\{x_i\}_{i=1}^n \subset \mathbb{R}^d$, let $\boldsymbol{\mu}$ and $\boldsymbol{S}$ be the sample mean and sample covariance matrix. The eigenvalue decomposition of $\boldsymbol{S}$ is $\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$, then using the first q eigenvactors we can project the data to a low dimensional space. P-PCA is to assume that the data are generated by

$$x = \boldsymbol{W}z + \boldsymbol{\mu} + \boldsymbol{\epsilon} \quad \text{where} \quad z \sim \mathcal{N}(0, I)\ ,\ \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 I),$$

$z \in \mathbb{R}^q$ and $\boldsymbol{W} \in \mathbb{R}^{d\times q}$. Then we have $x \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{W}\boldsymbol{W}^T + \sigma^2 I)$, $x|z \sim \mathcal{N}(\boldsymbol{W}z + \boldsymbol{\mu}, \sigma^2 I)$ and $z|x \sim \mathcal{N}(\boldsymbol{P}^{-1}\boldsymbol{W}^T(x-\boldsymbol{\mu}), \sigma^2\boldsymbol{P}^{-1})$ where $\boldsymbol{P} = \boldsymbol{W}^T\boldsymbol{W} + \sigma^2 I$. The maximum likelihood estimator of $\boldsymbol{W}$ and σ^2 are

$$\boldsymbol{W}_{\text{ML}} = \boldsymbol{U}_q(\boldsymbol{\Lambda}_q - \sigma^2 I)^{1/2} \quad \text{and} \quad \sigma^2_{\text{ML}} = \frac{1}{d-q}\sum_{i=q+1}^{d} \lambda_i,$$

where $\boldsymbol{U_q}$ is the matrix of the first q columns of $\boldsymbol{U}$, $\boldsymbol{\Lambda_q}$ is the matrix of the first q eigenvalues of $\boldsymbol{\Lambda}$. In the following study, we assume that n is large enough such that we can learn the true $\boldsymbol{\mu}_*$ and $\boldsymbol{\Sigma_*}$. Thus we have $\boldsymbol{S} = \boldsymbol{\Sigma}_*$, $\boldsymbol{U}_q = \boldsymbol{U}_{q*}$, $\boldsymbol{\Lambda}_q = \boldsymbol{\Lambda}_{q*}$ for the generative model.

4.2 Minimax problem of latent space adversarial training

To perturb the data in the latent space, Data will go through the encode-decode process $x \to z \to \Delta z + z \to x'$. Different sampling strategies may affect our analysis. To avoid this issue, we consider 3 strategies could be used in practice to craft adversarial examples.

Strategy 1 Sample $x \sim \mathcal{D}$, encode $z = \arg\max q(z|x) = \boldsymbol{P}^{-1}\boldsymbol{W}^T(x - \boldsymbol{\mu_*})$, add a perturbation Δz, and finally, decode $x_{adv} = \arg\max p(x|z + \Delta z) = \boldsymbol{W}(z + \Delta z) + \boldsymbol{\mu_*}$.

Strategy 2 Sample $x \sim \mathcal{D}$, then sample $z \sim q(z|x)$, add a perturbation Δz, and finally, sample $x_{adv} \sim p(x|z + \Delta z)$.

Strategy 3 Sample $z \sim \mathcal{N}(0, I)$, add a perturbation Δz, and then sample $x_{adv} \sim p(x|z + \Delta z)$. In this strategy, x_{adv} can be viewed as the adversarial example of $x = \arg\max_x q(z|x)$.

The following lemma shows that the adversarial examples can be unified in one formula. Hence sampling strategies will not affect our analysis.

Lemma 1 (Adversarial examples perturbed in the latent space)**.** *Using these 3 strategies, the adversarial examples can be unified as*

$$x_{adv} = x' + \boldsymbol{W}\Delta z \quad \textit{and} \quad x' \sim \mathcal{D}'_j = \mathcal{N}(\boldsymbol{\mu_*}, \boldsymbol{U_*}\boldsymbol{\Lambda}^{(j)}\boldsymbol{U_*}^T), \quad j = 1, 2, 3, \tag{5}$$

where

$$\boldsymbol{\Lambda}^{(1)} = \begin{bmatrix} (\boldsymbol{\Lambda_q} - \sigma^2 I)^2\boldsymbol{\Lambda_q}^{-1} & 0 \\ 0 & 0 \end{bmatrix} \quad \boldsymbol{\Lambda}^{(3)} = \begin{bmatrix} \boldsymbol{\Lambda_q} & 0 \\ 0 & \sigma^2 I \end{bmatrix}$$

$$\boldsymbol{\Lambda}^{(2)} = \begin{bmatrix} (\boldsymbol{\Lambda_q} - \sigma^2 I)^2\boldsymbol{\Lambda_q}^{-1} + (\boldsymbol{\Lambda_q} - \sigma^2 I)\boldsymbol{\Lambda_q}^{-1}\sigma^2 + \sigma^2 I & 0 \\ 0 & \sigma^2 I \end{bmatrix}.$$

If the data lie in a q dimensional subspace, i.e. the covariance matrix $\boldsymbol{\Sigma_*}$ is rank q, we have $\boldsymbol{\Lambda}^{(1)} = \boldsymbol{\Lambda}^{(2)} = \boldsymbol{\Lambda}^{(3)} = \boldsymbol{\Lambda_*}$. Then $\mathcal{D}' = \mathcal{D}$.

In general, the adversarial example can be decomposed into 2 parts, the change of distribution $x' \sim \mathcal{D}'$ and the small perturbation $\boldsymbol{W}\Delta z$. Therefore the adversarial expected risk can be written as the following minimax problem

$$\min_{\boldsymbol{\mu},\boldsymbol{\Sigma}} \mathcal{L}_{ls}(\boldsymbol{\mu},\boldsymbol{\Sigma};\mathcal{D}'_j) = \min_{\boldsymbol{\mu},\boldsymbol{\Sigma}} \mathbb{E}_{x'\sim\mathcal{D}'_j} \max_{\|\Delta z\|\leq\varepsilon} \ell(x'+\boldsymbol{W}\Delta z,\boldsymbol{\mu},\boldsymbol{\Sigma}), \quad j=1,2,3. \tag{6}$$

We aim to analyze the difference properties between the minimax problems in equations (4) and (6). We give the excess risk and optimal saddle point analysis in the following subsections. We leave all the proofs of the lemmas and theorems in appendix A.

4.3 Excess risk analysis

We consider the difference between $\mathcal{L}_{ls}$ and $\mathcal{L}$ given the true $\boldsymbol{\Theta}_*$, i.e. $\mathcal{L}_{ls}(\boldsymbol{\Theta}_*;\mathcal{D}'_j) - \mathcal{L}(\boldsymbol{\Theta}_*;\mathcal{D})$. It characterizes the excess risk incured by the optimal perturbation. To derive the expression of excess risk, we decompose it into two parts

$$\mathcal{L}_{ls}(\boldsymbol{\Theta}_*;\mathcal{D}'_j) - \mathcal{L}(\boldsymbol{\Theta}_*;\mathcal{D}) = \underbrace{\mathcal{L}_{ls}(\boldsymbol{\Theta}_*;\mathcal{D}'_j) - \mathcal{L}(\boldsymbol{\Theta}_*;\mathcal{D}'_j)}_{perturbation} + \underbrace{\mathcal{L}(\boldsymbol{\Theta}_*;\mathcal{D}'_j) - \mathcal{L}(\boldsymbol{\Theta}_*;\mathcal{D})}_{change\ of\ distribution}. \tag{7}$$

To simply the notation, we consider the Lagrange penalty form of the inner maximization problem in equation (6), i.e. $\max \ell(x'+\boldsymbol{W}\Delta z,\boldsymbol{\mu},\boldsymbol{\Sigma}) - L\|\Delta z\|^2/2$, where L is the Lagrange multiplier. The following theorem gives the solution in general case.

Theorem 2 (Excess risk)**.** *Let $\mathcal{L}_{ls}$ and $\mathcal{L}$ be the loss with and without perturbation in latent space (equations (6) and (3) respectively), given the non-robustly learned $\boldsymbol{\Theta}_* = (\boldsymbol{\mu}_*,\boldsymbol{\Sigma}_*)$, The excess risk caused by perturbation is*

$$\mathcal{L}_{ls}(\boldsymbol{\Theta}_*,\mathcal{D}'_j) - \mathcal{L}(\boldsymbol{\Theta}_*,\mathcal{D}'_j) = \frac{1}{2}\sum_{i=1}^{q}\Big[(1+\frac{\lambda_i-\sigma^2}{(L-1)\lambda_i+\sigma^2})^2-1\Big]\frac{\lambda_i^{(j)}}{\lambda_i}, \quad j=1,2,3.$$

and the excess risk caused by changed of distribution is

$$\mathcal{L}(\boldsymbol{\Theta}_*,\mathcal{D}'_j) - \mathcal{L}(\boldsymbol{\Theta}_*,\mathcal{D}) = \frac{1}{2}\log\Big[\frac{\prod_{i=1}^d \lambda_i^{(j)}}{\prod_{i=1}^d \lambda_i}\Big] + \frac{1}{2}\big(\sum_{i=1}^d \frac{\lambda_i^{(j)}}{\lambda_i} - d\big).$$

It is hard to see which part dominates the excess risk. If we further assume that the data lie in a q dimensional manifold. The excess risk caused by the change of distribution becomes 0 by Lemma 1. We have the following corollary.

Corollary 3 (Excess risk)**.** *Let $\mathcal{L}_{ls}$ and $\mathcal{L}$ be the loss with or without perturbation in latent space (equation (6) and (3) respectively), given the non-robustly learned $\boldsymbol{\Theta}_* = (\boldsymbol{\mu}_*,\boldsymbol{\Sigma}_*)$, and rank$(\boldsymbol{\Sigma}_*) = q$. The excess risk*

$$\mathcal{L}_{ls}(\boldsymbol{\Theta}_*,\mathcal{D}'_j) - \mathcal{L}(\boldsymbol{\Theta}_*,\mathcal{D}) = \mathcal{O}(qL^{-2}).$$

We compare it with the excess risk of original space adversarial training.

Theorem 4 (Excess risk of orginal space adversarial training)**.** *Let $\mathcal{L}_r$ and $\mathcal{L}$ be the loss with or without perturbation in original space (equations (4) and (3) respectively), given the non-robustly learned $\boldsymbol{\Theta}_* = (\boldsymbol{\mu}_*,\boldsymbol{\Sigma}_*)$. Denote λ_{min} be the smallest eigenvalue of $\boldsymbol{\Sigma}_*$. The excess risk*

$$\Omega((\lambda_{min}L)^{-2}) \leq \mathcal{L}_r(\boldsymbol{\Theta}_*,\mathcal{D}) - \mathcal{L}(\boldsymbol{\Theta}_*,\mathcal{D}) \leq \mathcal{O}(d(\lambda_{min}L)^{-2}).$$

Comparison of excess risk Compare the difference between Corollary 3 and Theorem 4, we can see that optimal perturbation in the latent space will inccur an excess risk in $\mathcal{O}(qL^{-2})$. The adversarial vulnerability comes from the dimension q. Generative Attacks focus on the largest q variance directions. Optimal perturbation in the original space will inccur an excess risk at least in $\Omega((\lambda_{min}L)^{-2})$. The adversarial vulnerability comes from the smallest eigenvalues λ_{min}. Regular Attacks focus on small variance directions.

4.4 SADDLE POINT ANALYSIS

In this subsection we study the optimal solution of optimization problem (6). Since it is not a standard minimax problem, we consider a modified problem.

$$\min_{\boldsymbol{\mu},\boldsymbol{\Sigma}} \max_{\mathbb{E}_{x'}\|\Delta z\|=\varepsilon} \mathbb{E}_{x'\sim\mathcal{D}'_j} \ell(x' + \boldsymbol{W}\Delta z, \boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad j = 1, 2, 3. \tag{8}$$

We will explain more about the connection between optimization problems in equation (6) and (8). See appendix A. The following theorem is our main result. It gives the optimal solution of Latent space adversarial training.

Theorem 5 (Main result: Optimal Saddle point)**.** *The optimal solution of the modified problem in equation (8) is*

$$\boldsymbol{\mu}_{ls} = \boldsymbol{\mu}_* \quad \textit{and} \quad \boldsymbol{\Sigma}_{ls} = \boldsymbol{U}_* \boldsymbol{\Lambda}^{ls} \boldsymbol{U}_*^T,$$

where

$$\lambda_i^{ls} = \frac{1}{4}\Big[2\lambda_i^{(j)} + \frac{4(\lambda_i - \sigma^2)}{L} + 2\lambda_i^{(j)}\sqrt{1 + \frac{4(\lambda_i - \sigma^2)}{\lambda_i^{(j)} L}}\Big] \textit{ for } i = 1 \le q \textit{ and } \lambda_i^{ls} = \lambda_i^{(j)} \textit{ for } i > q.$$

$j = 1, 2, 3$ *corresponding to strategies 1,2 and 3.*

We assume that the data lie in a q dimension manifold again. Then we have $\lambda_i^{ls}/\lambda_i = 1/2 + 1/L + \sqrt{1/4 + 1/L} \ge 1$ for $i \le q$. Latent space adversarial training increases the model robustness by **amplifying large eigenvalues** of the data manifold. The two dimensional case is illustrated in Figure 1, (c) and (d).

In the same setting, the optimal solution of standard adversarial training (Problem (4)) on the original space is in Theorem 6 (which is the Theorem 2 in Ilyas et al. (2019)).

Theorem 6 (Optimal saddle point Ilyas et al. (2019))**.** *The optimal solution of problem in equation (4) is*

$$\boldsymbol{\mu}_r = \boldsymbol{\mu}_* \quad \textit{and} \quad \boldsymbol{\Sigma}_r = \frac{1}{2}\boldsymbol{\Sigma}_* + \frac{1}{L}I + \sqrt{\frac{1}{L}\boldsymbol{\Sigma}_* + \frac{1}{4}\boldsymbol{\Sigma}_*^2}.$$

Theorem 6 is for the problem that the covariance matrix is restricted to be diagonal. Consider the ratio

$$\frac{\lambda_i^{(r)}}{\lambda_i} = \frac{1}{2} + \frac{1}{L\lambda_i} + \sqrt{\frac{1}{L\lambda_i} + \frac{1}{4}}.$$

For small true eigenvalue λ_i, the ratio is large. Standard adversarial training increases the robustness by **amplifying the small eigenvalues** of the data manifold. The two dimensional case is illustrated in Figure 1, (a) and (b).

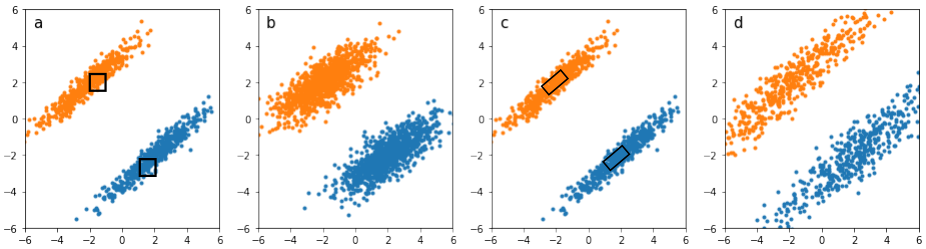

Figure 1: Demonstration of theoretical analysis: (a) Regular attacks directions; (b) Optimal saddle point of original space adversarial training; (c) Generative attacks directions; (d) Optimal saddle point of latent space adversarial training.

5 EXPERIMENTS

In this section we report experimental results on training Lenet on Mnist LeCun et al. (1998) and resnet He et al. (2016) on CIFAR-10Krizhevsky et al. (2009) to confirm our theoretical findings. Details of the hyperparameters setting are in appendix B.

5.1 EIGENVALUES OF MNIST WITH AND WITHOUT ADVERSARIAL TRAINING

In this subsection, we show that the properties of the eigenvalues are also valid on the real dataset. We use MNIST as an example. Firstly, for each image in the test set $\mathcal{D}_{test}$ of MNIST, we vectorize it into a 784 dimension vector. Then we calculate the EVD (Eigenvalue decomposition) of the covariance matrix of each class. We plot the 784 eigenvalues of class 0 and 1 in Figure 2 in blue line. Secondly, we adversarially train a classifier $f_r(x)$ in original space. For each point x in $\mathcal{D}_{test}$, we construct x_r by PGD-attack (which is called norm-based attack in Figure 2) with $\nabla_x f_r(x)$. Then we obtain a robust test set $\mathcal{D}^r_{test}$. We plot the intra class eigenvalues of the covariance matrix in Figure 2 in orange line. Lastly, we adversarially train a classifier $f_{ls}(x)$ in latent space. Then we construct latent space robust dataset $\mathcal{D}^{ls}_{test}$ by generative attack against $f_{ls}(x)$ on $\mathcal{D}_{test}$. The eigenvalues of the covariance matrix are shown in Figure 2 in green line. We show all the eigenvalues in the first column in Figure

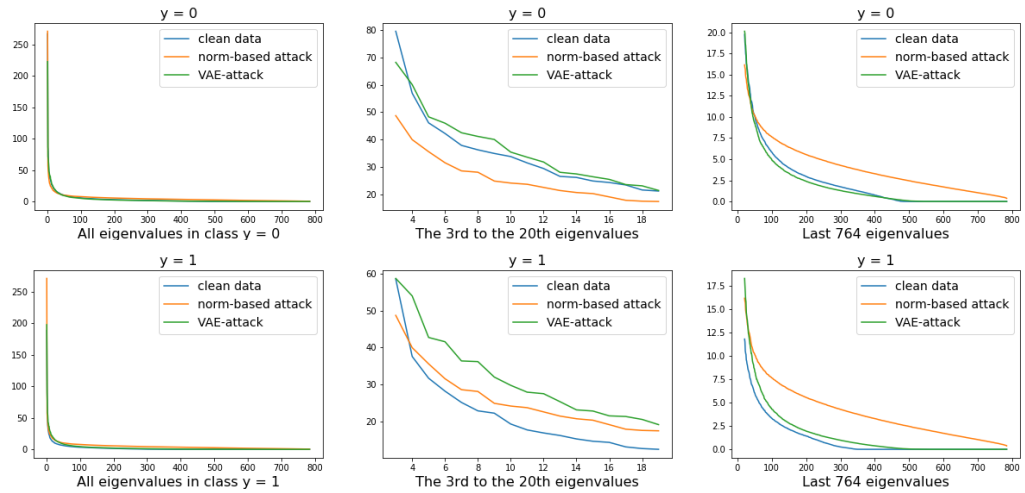

Figure 2: Eigenvalues of MNIST: The first row and the second row are of classes 0 and 1 respectively. The first column is about all the 784 eigenvalues of the dataset. The second column plots the large eigenvalues and the last column plots the small eigenvalues.

2. After adding a small perturbation in the original space or in the latent space, the distribution of $\mathcal{D}^r_{test}$ and $\mathcal{D}^{ls}_{test}$ are closed to $\mathcal{D}_{test}$. Because the first two eigenvalues are too large, the values of the y-axis are too large to distinguish the difference among these 3 distributions. We omit the top two eigenvalues and look into the details of large eigenvalues (the 3rd to the 20th eigenvalues) in the second column and small eigenvalues (last 764 eigenvalues) in the last column.

Original space adversarial training focuses on the small variance direction We adversarially train a robust classifier $f_r(x)$ in the original space. The robust test set $\mathcal{D}^r_{test}$ against $f_r(x)$ amplified the small eigenvalues a lot. In the third column, the orange line is significantly larger that the other 2 lines. While in the second column, the orange line may or may not amplified the large eigenvalues. The experiments give us an understanding of how the adversarial examples leave the data manifolds. The original dataset $\mathcal{D}_{test}$ lies in a low dimension affine plane in $\mathbb{R}^{784}$. after adversarial training, the data move towards the small variance directions. Finally $\mathcal{D}^r_{test}$ stay in a full dimension subset of $\mathbb{R}^{784}$.

Latent space adversarial training focuses on the large variance direction We adversarially train a robust classifier $f_{ls}(x)$ in the latent space. The latent space robust test set $\mathcal{D}^{ls}_{test}$ against $f_{ls}(x)$ amplified the large eigenvalues. In the second column, we can see that the green line is above the other two lines. And in the last column, the green line shows that adversarial training in the latent space will not affect the small eigenvalues. The adversarial examples in $\mathcal{D}^{ls}_{test}$ move along the large variance direction. Therefore, $\mathcal{D}^{ls}_{test}$ stay in the low dimension affine space.

5.2 ROBUST TEST ACCURACIES

In this subsection, we compare the test accuracy of different attacks (FGSM, PGD, VAE) versus different defense (adversarial training with FGSM, PGD, and VAE) to help explain our theoretical results. We explain the results of MNIST in Figure 1 as an example.

On-manifold and off-manifold adversarial examples The test accuracies of the standard training model on PGD and VAE-attack data are 3.9% and 42.4% respectively. Firstly, They show that on-manifold adversarial examples exist. Then, they support that PGD-adversarial examples move towards low variance directions (leave the low dimension manifold), while Gen-adversarial examples move towards high variance directions (move within the manifold).

Attack versus defense Our theory tells us that after original or latent space adversarial training, the model robustness is increased by amplifying the small or large variance in the data manifold respectively. In experiments, test accuracy of VAE-adv vs PGD-attack is 1.23%, which shows that amplifying high variance of distribution gives no contribution to defending low variance directions attacks. Similarly, the test accuracy of PGD-adv vs VAE-attack is 52.18%, which shows that amplifying low variance of distribution gives no contribution to defending high variance direction attacks.

Training with 2 types of adversarial examples Our theory suggests that adversarial robustness can be disentangled in different directions. Hence adversarial robustness against attacks on low variance and large variance can be guaranteed simultaneously. In the last column of Table 1, we show the results of using both regular adversarial examples and generative adversarial examples to train the model. We can see that we can increase the model robustness against all attacks. The results on

Mnist	Std training	FGSM-Adv	PGD-Adv	VAE-Adv	**Both-Adv**
clean data	98.82%	98.79%	98.73%	98.28%	**98.80%**
FGSM-Attack	47.43%	91.08%	96.50%	27.94%	**96.54%**
PGD-Attack	3.90%	15.46%	95.51%	1.23%	**89.50%**
VAE-Attack	42.40%	46.94%	52.18%	96.66%	**90.28%**
CIFAR-10	Std training	FGSM-Adv	PGD-Adv	VAE-Adv	**Both-Adv**
clean data	92.14%	87.03%	86.25%	85.59%	**85.80%**
FGSM-Attack	38.39%	76.92%	77.34%	34.50%	**76.17%**
PGD-Attack	7.40%	75.14%	75.78%	10.53%	**74.22%**
VAE-Attack	19.40%%	31.80%	26.31%	40.18%	**42.47%**

Table 1: Test accuracy of different attacks and defense algorithms on Mnist and CIFAR-10

CIFAR-10 are not as significant as the results on MNIST. Possible reasons are the modeling capacity of vanilla VAE (which will be discussed in appendix B). But it still confirms our findings.

6 CONCLUSION

In this paper, we study the problem of adversarial robustness against on-manifold and off-manifold adversarial examples. We show that adversarial robustness can be disentangled in small variance directions and large variance directions of the data manifold. Theoretically, we study the excess risk and optimal saddle point of the minimax problem of latent space adversarial training. The optimal solution shows that latent space adversarial training increases model robustness by amplifying the large variance of the distribution. On the other hand, original space adversarial training increases model robustness by amplifying the small variance of the distribution. Experimentally, we show that these phenomena also exist in real datasets. It gives us a new perspective to understand the behavior of on-manifold and off-manifold adversarial examples. In future works, we can further analyze the conflict between robustness and generalization, which is an open problem, based on this work.

REFERENCES

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

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.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014a.

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, 2014b.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. *arXiv preprint arXiv:1710.11342*, 2017.

Yang Song, Rui Shu, Nate Kushman, and Stefano Ermon. Generative adversarial examples. *arXiv preprint arXiv:1805.07894*, 2018a.

Jernej Kos, Ian Fischer, and Dawn Song. Adversarial examples for generative models. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 36–42. IEEE, 2018.

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, 2018b.

David Stutz, Matthias Hein, and Bernt Schiele. Disentangling adversarial robustness and generalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6976–6987, 2019.

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.

Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John C Duchi, and Percy Liang. Adversarial training can hurt generalization. *arXiv preprint arXiv:1906.06032*, 2019.

Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26, 2017.

Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. *arXiv preprint arXiv:1807.07978*, 2018.

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.

Lukas Schott, Jonas Rauber, Matthias Bethge, and Wieland Brendel. Towards the first adversarially robust neural network model on mnist. *arXiv preprint arXiv:1805.09190*, 2018.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*, 2018.

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.

Vishaal Munusamy Kabilan, Brandon Morris, Hoang-Phuong Nguyen, and Anh Nguyen. Vectordefense: Vectorization as a defense to adversarial examples. In *Soft Computing for Biomedical Applications and Related Topics*, pages 19–35. Springer, 2018.

Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. Deflecting adversarial attacks with pixel deflection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8571–8580, 2018.

Guneet S Dhillon, Kamyar Azizzadenesheli, Zachary C Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. Stochastic activation pruning for robust adversarial defense. *arXiv preprint arXiv:1803.01442*, 2018.

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 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.

Ajil Jalal, Andrew Ilyas, Constantinos Daskalakis, and Alexandros G Dimakis. The robust manifold defense: Adversarial training using generative models. *arXiv preprint arXiv:1712.09196*, 2017.

Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pages 3483–3491, 2015.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pages 125–136, 2019.

Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

Soheil Feizi, Farzan Farnia, Tony Ginart, and David Tse. Understanding gans: the lqg setting. *arXiv preprint arXiv:1710.10793*, 2017.

Soheil Feizi, Farzan Farnia, Tony Ginart, and David Tse. Understanding gans in the lqg setting: Formulation, generalization and stability. *IEEE Journal on Selected Areas in Information Theory*, 2020.

Jaewoong Cho and Changho Suh. Wasserstein gan can perform pca. In *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 895–901. IEEE, 2019.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Bin Dai, Yu Wang, John Aston, Gang Hua, and David Wipf. Connections with robust pca and the role of emergent sparsity in variational autoencoder models. *The Journal of Machine Learning Research*, 19(1): 1573–1614, 2018.

Bin Dai, Yu Wang, John Aston, Gang Hua, and David Wipf. Hidden talents of the variational autoencoder. *arXiv preprint arXiv:1706.05148*, 2017.

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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Constantinos Daskalakis, Themis Gouleakis, Chistos Tzamos, and Manolis Zampetakis. Efficient statistics, in high dimensions, from truncated samples. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 639–649. IEEE, 2018.