# ADVERSARIAL LEARNING ON ROBUSTNESS AND GENERATIVE MODELS

by

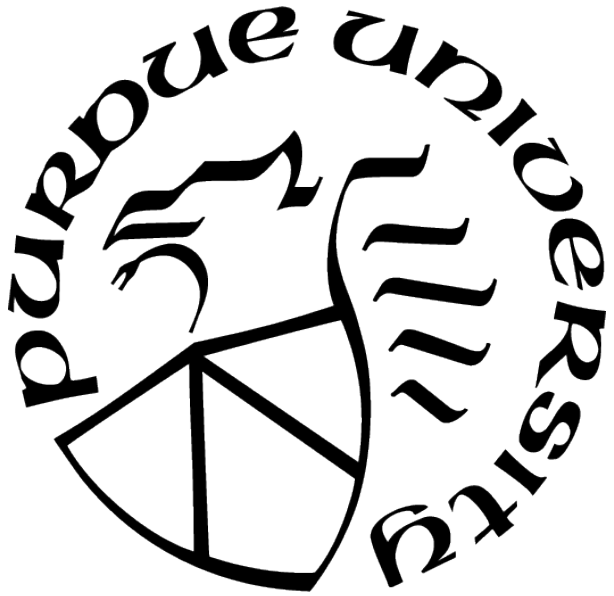
**Qingyi Gao**

**A Dissertation**

*Submitted to the Faculty of Purdue University*

*In Partial Fulfillment of the Requirements for the degree of*

**Doctor of Philosophy**




Department of Statistics

West Lafayette, Indiana

August 2021

# THE PURDUE UNIVERSITY GRADUATE SCHOOL
# STATEMENT OF COMMITTEE APPROVAL

**Dr. Xiao Wang, Chair**

Department of Statistics, Purdue University

**Dr. Hyonho Chun, Co-Chair**

Department of Mathematical Sciences, Korea Advanced Institute of Science and Technology

**Dr. Guang Cheng**

Department of Statistics, Purdue University

**Dr. Jun Xie**

Department of Statistics, Purdue University

**Approved by:**

Dr. Jun Xie

To my family.

# ACKNOWLEDGMENTS

Throughout the academic journey these years, I have received a great deal of supports and helps from the department of statistics at Purdue University.

First and foremost, I would like to express my sincere and deepest gratitude towards my esteemed advisor, Dr. Xiao Wang, who led me to join the fantastic and mysterious deep learning field. Dr. Xiao Wang has provided me many invaluable supports and guidance during my PhD study. His patience and enthusiasm on research motivate me to catch up with the updated developments in this field. I have learned a lot from him, such as independent and critical thinking, rigorous attitude for research and scientific writing.

Besides my advisor, I would like to thank my other three committee members, Dr. Hyonho Chun, Dr. Guang Cheng, and Dr. Jun Xie, for their insightful comments and questions that help me complete my dissertation. Especially, I am extremely grateful to my co-advisor Dr. Hyonho Chun for the corporation with Dr. Hubo Cai from the School of Civil Engineering, from which I learnt how to efficiently and neatly present work to people with different backgrounds.

I appreciate to work in STAT 301 team as a TA coordinator. Dr. Laura Cayon gave me many useful guidance and suggestions on running the TA sessions especially during the pandemic year.

My sincere thanks also go to my fellow students that I spent with at Purdue, Yixuan Qiu, Yixi Xu, Sophie Sun, Yao Chen, Jiapeng Liu, Jungeum Kim, Yijia Liu, Siqi Liang, Tianning Dong, Xiaochen Yang, Bingjing Tang, Haoyun Yin, Huiming Xie and Chuanhui Liu. I especially thank my best friends Botao Hao and Peiyi Zhang, for their unlimited helps and all the fun we have had during the past five years.

Last but not the least, I would like to thank my family for their unconditional love and supports.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF SYMBOLS

| | |
|---|---|
| bold-faced letter | vector, e.g., $\mathbf{x} = (x_1, \ldots, x_{p_0})$ and $\mathbf{z} = (z_1, \ldots, z_{d_0})$ |
| capital letter | matrix, e.g., $W$ and $A$ |
| $[L]$ | set $\{1, 2, \ldots, L\}$ |
| $\|\mathbf{x}\|_p$ | $\ell_p$-norm of a vector $\mathbf{x}$ for $p > 0$, e.g., $\|\mathbf{x}\|_p = (\sum_{\mathrm{i}=1}^{p_0} \|x_{\mathrm{i}}\|^p)^{1/p}$. When $p = 2$, we may ignore its subscript and denote $\|\mathbf{x}\|$. |
| $\|W\|_{p,q}$ | $L_{p,q}$-norm of a matrix $W$ for $p > 0$ and $q > 0$, e.g., $\|W\|_{p,q} = \Big( \sum_{\mathrm{j}} (\sum_{\mathrm{i}} \|W_{\mathrm{i,j}}\|^p)^{q/p} \Big)^{1/q}$ |
| $\|W\|_F$ | Frobenius norm of a matrix $W$, which equals to $\|W\|_{2,2}$. |
| $\|W\|_2$ | spectral norm of a matrix $W$, which equals to its largest singular value. |
| $\|W\|_*$ | nuclear norm of a matrix $W$, which equals to the sum of its singular values. |

# ABBREVIATIONS

| | |
|---|---|
| CelebA | CelebFaces Attributes Datase |
| CNN | convolutional neural network |
| DNN | deep neural network |
| ELBO | evidence lower bound |
| FGSM | fast gradient sign method |
| FID | Frechet inception distance |
| GAN | generative adversarial network |
| IS | inception score |
| Isomap | isometric mapping |
| KL divergence | Kullback-Leibler divergence |
| LLE | locally linear embedding |
| LWGAN | latent Wasserstein generative adversarial network |
| MCMC | Markov chain Monte Carlo |
| MDS | multi-dimensional scaling |
| MLE | maximum likelihood estimation |
| MMD | maximum mean discrepency |
| MNIST | Modified National Institute of Standards and Technology dataset |
| PGD | projected gradient descent |
| RE | reconstruction error |
| SVD | singular value decomposition |
| VAE | variational auto-encoder |
| WAE | Wasserstein auto-encoder |
| WGAN | Wasserstein generative adversaril network |

# ABSTRACT

In this dissertation, we study two important problems in the area of modern deep learning: adversarial robustness and adversarial generative model. In the first part, we study the generalization performance of deep neural networks (DNNs) in adversarial learning. Recent studies have shown that many machine learning models are vulnerable to adversarial attacks, but much remains unknown concerning its generalization error in this scenario. We focus on the $\ell_\infty$ adversarial attacks produced under the fast gradient sign method (FGSM). We establish a tight bound for the adversarial Rademacher complexity of DNNs based on both spectral norms and ranks of weight matrices. The spectral norm and rank constraints imply that this class of networks can be realized as a subset of the class of a shallow network composed with a low dimensional Lipschitz continuous function. This crucial observation leads to a bound that improves the dependence on the network width compared to previous works and achieves depth independence. We show that adversarial Rademacher complexity is always larger than its natural counterpart, but the effect of adversarial perturbations can be limited under our weight normalization framework.

In the second part, we study deep generative models that receive great success in many fields. It is well-known that the complex data usually does not populate its ambient Euclidean space but resides in a lower-dimensional manifold instead. Thus, misspecifying the latent dimension in generative models will result in a mismatch of latent representations and poor generative qualities. To address these problems, we propose a novel framework called Latent Wasserstein GAN (LWGAN) to fuse the auto-encoder and WGAN such that the intrinsic dimension of data manifold can be adaptively learned by an informative latent distribution. In particular, we show that there exist an encoder network and a generator network in such a way that the intrinsic dimension of the learned encodes distribution is equal to the dimension of the data manifold. Theoretically, we prove the consistency of the estimation for the intrinsic dimension of the data manifold and derive a generalization error bound for LWGAN. Comprehensive empirical experiments verify our framework and show that LWGAN is able to identify the correct intrinsic dimension under several scenarios, and simultaneously generate high-quality synthetic data by samples from the learned latent distribution.

# 1. INTRODUCTION

In modern machine learning and statistics, there are two main types of tasks: supervised learning and unsupervised learning. The goal of supervised learning is to learn a function that best approximates the relationship between input and observable output in the data. Unsupervised learning, on the other hand, does not have labeled outputs, so its goal is to infer the natural structure present within a set of data points. These are the essential and fundamental problem in many complex scenarios. For example, providing the the words in a document (input) and its corresponding topic (output), we seek a mapping $f(\cdot)$ that can classify future document correctly. On the other hand, if we are trying to segment consumers, unsupervised clustering methods would be a great starting point.

Deep neural networks (DNNs), one of the powerful function approximators in recent decades, have made significant progresses for the problem of learning from data in both supervised and unsupervised tasks. In particular, DNNs have demonstrated an amazing performance in solving complicated artificial intelligence tasks such as image generation, object recognition and identification, text understanding and translation, and many other domains [1]. They have become popular due to their predictive power and flexibility in model fitting. However, from both theoretical and applicant point of views, several essential issues have aroused.

In the first part of the dissertation, we focus on the supervised learning. Many studies have shown that DNNs are vulnerable to adversarial attacks [2]–[4]. The adversarial inputs are called *adversarial examples*, which are typically generated by adding small perturbations that are imperceptible to human eyes [5] to the original data. Formally, if we take an example $\mathbf{x} \in \mathbb{R}^p$ belonging to the class $c_1$ as input, there are several efficient algorithms to find the adversarial example $\mathbf{x}'$ such that $\mathbf{x}'$ is very close to $\mathbf{x}$ but the classifiers incorrectly predict it as belonging to class $c_2 \neq c_1$. Other methods of generating adversarial examples include rotation and translation [6] and background changing [7]. Many deep learning models achieve state-of-the-art performance in benchmark datasets, but they perform poorly on these adversarial examples. For example, the adversarial test accuracy on CIFAR10 is reported as only 47% in [8], instead the natural test accuracy on CIFAR10 is around 95% [9]. Here CIFAR10

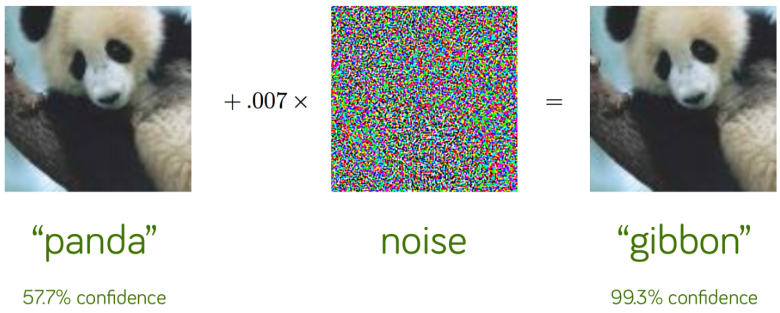



**Figure 1.1.** A demonstration of fast adversarial example generation applied to GoogLeNet on ImageNet. By adding an imperceptibly small vector whose elements are equal to the sign of the elements of the gradient of the cost function with respect to the input, we can change GoogLeNet's classification of the image.

is a benchmark dataset which consists of 60,000 $32 \times 32$ color images in 10 classes [10]. Particularly, in the area of cybersecurity, learning models face adversaries that try to deceive learning models and avoid being detected. The notorious example is *DeepFake* [11], which is an AI-based technology used to produce or alter video content so that it presents something that did not, in fact, occur.

Recently, there has been much progress towards the development of models achieving robustness [8], [12]–[15] through the *adversarial learning*, which is a technique that attempts to fool models by supplying deceptive input. Suppose that the sample $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$ is drawn according to some unknown underlying distribution $P$, where $\mathcal{X}$ and $\mathcal{Y}$ are input and label domains respectively. Let $\mathcal{F}$ be a class of DNNs with a particular architecture and $g(f(\mathbf{x}), y)$ be the loss function associated with $f \in \mathcal{F}$. Then the adversarially robust model is learned by minimizing the *empirical adversarial risk*, that is,

$$\min_{f \in \mathcal{F}} \frac{1}{m} \sum_{\mathrm{i}=1}^{m} \max_{\|\mathbf{x}'_{\mathrm{i}} - \mathbf{x}_{\mathrm{i}}\|_p \leq \epsilon} g(f(\mathbf{x}'_{\mathrm{i}}), y_{\mathrm{i}}). \tag{1.1}$$

Suppose $\tilde{\mathbf{x}}_{\mathrm{i}}^{*}$s, for $\mathrm{i} = 1, \ldots, m$, are the optimal solution to the inner maximize problem, then the generalization error in the *adversarial setting* is

$$\mathbb{E}_{(\tilde{\mathbf{x}}^{*}, y)}[g(f(\tilde{\mathbf{x}}^{*}), y)] - \frac{1}{m}\sum_{\mathrm{i}=1}^{m} g(f(\tilde{\mathbf{x}}_{\mathrm{i}}^{*}), y_{\mathrm{i}}). \tag{1.2}$$

Adversarial generalization behavior measures how accurately a model is able to predict outcome values for previously unseen data, and a small value is expected to avoid over-fitting. Unfortunately, Equation 1.1 is an intractable optimization problem for DNNs, so several adversarial strategies are proposed to approximate $\tilde{\mathbf{x}}^{*}$ such as Fast Gradient Method (FGM) [16] and Projected Gradient Descent (PGD) [8]. These methods produce white-box attacks where the attacker has access to the model's parameters, which implies that the attack relies on the model $f$.

However, [17] shows that the performance of adversarially trained DNNs over test samples can be significantly worse than their training performance, and this gap can be far greater than the generalization gap achieved in the *natural setting*, i.e., inputs without adversarial attacks. To close the discrepancy, [18] applies the spectral-norm regularization during adversarial training, and extends PAC-Bayes framework to bound the generalization error for DNNs under FGM and PGD, but they only focus on $\ell_2$ attacks. When $\ell_\infty$-norm attacks and multi-class linear classifiers are considered, [19] establishes an adversarial generalization bound depending on the number of class $K$ and the dimension of input $p$ with an order of $\sqrt{K^3 p}$ if $L_{2,\infty}$-norm constraint is adapted. [20] derives the surrogate risk bound relying on $\ell_\infty$-operator norms, Frobenius norms, and $L_{p,\infty}$ norms of weight matrices, which is polynomially depend on the depth of DNNs. Overall, existing works only consider simple models or depend on the size of the network, so we further investigate the generalization property of $\ell_\infty$ attack on DNNs to theoretically gain deeper understanding of this problem. For example, whether the generalization error under the $\ell_\infty$ attack could be size-free given some assumptions.

In this part of the dissertation, we provide a tighter sample complexity bounds for adversarially robust generalization of DNNs based on both spectral norms and ranks of weight matrices under the $\ell_\infty$ adversarial attacks. We compare the adversarial Rademacher com-

plexity with the natural Rademacher complexity. The adversarial complexity is never smaller than its natural counterpart, but the effect of adversarial perturbations can be limited under our weight normalization framework. We further conduct experiments on neural networks with different depth to verify our theoretical findings.

In the second part of my dissertation, we study deep generative model in the context of unsupervised learning. In the past few years, deep learning based generative models have gained a lot of interest due to the amazing improvements in the field [21]–[26]. Leveraging huge amount of data, well-designed networks architectures and advanced training techniques, deep generative models have shown an incredible ability to produce highly realistic pieces of content of various kind, such as images, texts and sounds. Given a random sample $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^p$ drawn from an unknown distribution $P_X$, the goal is to train a generative model that can produce synthetic data points that look similar to the real data. While there are several ways to quantify the similarity, the most common approach is to directly employ some of the known divergence measures, such as the Kullback-Leibler divergence and the Wasserstein distance, between the real data distribution and synthetic data distribution.

There are three major frameworks for generative models: Variational Auto-Encoders (VAEs) [21], Generative Adversarial Networks (GANs) [22], and Normalizing Flows [25]. The generative models are usually latent variable models through a latent variable $Z \in \mathcal{Z} \subset \mathbb{R}^d$ drawn from a simple and accessible prior distribution $P_Z$, such as $P_Z = N(0, I)$. Then, the synthetic data are generated by either a deterministic transformation $G(\mathbf{z})$ or a conditional distribution $p(\mathbf{x}|\mathbf{z})$. In particular, VAEs maximize the lower bound of the log likelihood $\log p_\theta(\mathbf{x})$, so they have strong theoretical justifications and typically can cover all modes of the data distribution. However, they often produce blurry images. GANs simultaneously learn a generator and a discriminator by pushing the powerful discriminator to distinguish between real data and generative samples. GANs can generate visually realistic images, but suffer from unstable training and mode collapsing. Normalizing Flows learn a generative model by directly maximizing the exact log-likelihood of well-specified probabilistic models, so they require the dimension of the latent space to be identical to that of the original data space. This results in an invertible generator and a tractable density that inhabits

the full data space. However, high-dimensional latent space usually prohibits an efficient representation learning.

There are several limitations for the above generative models. It is a requirement for current approaches of training generative models to pre-specify the dimension of the latent distribution $P_Z$ and treat it as fixed during the training process. For example, the latent dimensions for VAEs and GANs are pre-specified by users, and the latent dimension for Normalizing Flows is instead kept the same as the dimension of the data. It is known that many observed data such as natural images lie on a low-dimensional manifold embedded in a higher dimensional space. Therefore, an inappropriate choice of the latent dimension could lead to a wrong latent representation that does not populate the full ambient space [27]. The wrongly specified latent dimension will fail to uncover the structure of the data. Overall, the corresponding generative models may suffer from mode collapse, under-fitting, mismatch of representation learning, and poor generative qualities. Furthermore, some fundamental divergences such as Maximum Likelihood Estimation (MLE) and KL divergence are ill-defined that brings additional challenges for model training. Although there are many interesting works take advantages of both VAEs and GANs [28]–[30], it remains unclear what the principles are underlying the framework combining the best of WAEs and WGANs.

To handle the aforementioned drawbacks, we propose a novel approach, called Latent Wasserstein GAN (LWGAN), to identify the intrinsic dimension of a data distribution that lies on a continuous manifold. This approach could greatly improve the quality of generative modeling as well as representation learning. Specifically, to learn a informative prior distribution $P_Z$, we utilize a deterministic encoder $Q$ borrowed from the WAE. On the other hand, a generator $G$ is combined to generate images from the latent code $Z \sim P_Z$ that look like the real ones. To get rid of possible invalid divergences, we focus on the 1-Wasserstein distance to measure the similarities between two distributions, which apply to any two distributions as long as they can be sufficiently sampled. After training, the estimated intrinsic dimension of the prior distribution $P_Z$ consists with the true intrinsic dimension. We conduct comprehensive experiments to confirm that LWGAN is able to detect the correct intrinsic dimension under several settings using both toy example as well as real data such as MNIST and CelebA.

## 1.1 Dissertation Organisation

The remaining parts of the dissertation is organized as follows. In chapter 2, we provide a brief view of some neural network architectures, including the basic feed-forward neural networks as well as three major types of generative models such as VAEs, GANs and Normalizing Flows. In chapter 3, we introduce the fundamental techniques of statistical learning theory, whose goal is to control the difference between population risk and empirical risk, a.k.a, generalization error. For example, the Rademacher complexity and VC dimension are used to describe the capacity of a class of functions, which are the links to the upper bound of generalization error.

In chapter 4, we establish our first contribution on the theoretical study of adversarial learning. Section 4.1 describes the role of spectral norm and low-rank weight matrix in DNN. Section 4.2 establishes the generalization bounds for adversarial learning. This includes a regular upper bound and a tight upper bound on Rademacher complexities for adversarial learning. Section 4.3 compares the generalization behaviors between natural learning and adversarial learning. Numerical results are provided in Section 4.4 to validate our theoretical conclusions. Section 4.5 gives the proofs of our theories.

Starting from chapter 5, we switch to our second contribution, which studies the adversarial generative models on the latent space. Section 5.1 investigates the phenomenon of dimensional mismatch between the latent distribution and data distribution, and establishes the existence of an encoder and generator that can solve the mismatch dilemma. Section 5.2 presents the new LWGAN framework that provides a feasible way of obtaining the mentioned encoder and generator. Theoretical guarantees are given in Section 5.3, including rank consistency and generalization error bounds. Section 5.4 demonstrates extensive numerical experiments under different settings to verify that LWGAN is able to detect the correct intrinsic dimension for both toy examples and natural data such as MNIST and CelebA. Section 5.5 provides related proofs of this chapter.

Chapter 6 is a summary of the dissertation and some future works.

# 2. PRELIMINARIES

## 2.1 Feed-forward Neural Networks

Feed-forward neural networks, also called multilayer perceptrons, are the quintessential deep learning models. The goal of a feed-forward neural network is to approximate some true functions. [31] stated that a feedforward network with at least one hidden layer with any "squashing" activation function (such as the logistic sigmoid activation function) can approximate any Borel measurable function from one finite-dimensional space to another with any desired non-zero amount of error, provided that the network is given enough hidden units. This universal property makes it the basis of many important commercial applications. For instance, the convolutional neural networks used for object recognition from photos are a specialized kind of feedforward network. It is also a conceptual stepping stone on the path to recurrent networks, which power many natural language applications.

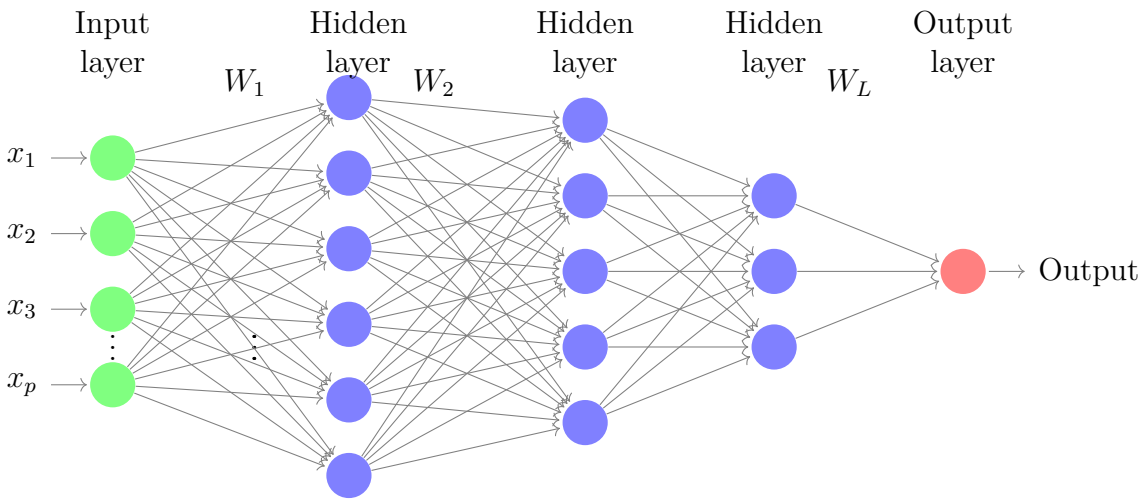



**Figure 2.1.** Deep Neural Networks

The architecture of neural networks is shown in Figure 2.1, which is composed of a input layer, multiple hidden layers and a output layer. Each layer contains one or more neurons, the basic block of a network. The total number of layers is called the depth of a network, and the maximum number of neurons in each layer is the width of the network. Mathematically,

given the input domain $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^p\}$, we use $f_{\mathcal{W}_L}(\mathbf{x})$ to denote a neural networks of the form

$$f_{\mathcal{W}_L}(\mathbf{x}) = T_L \circ \sigma_L \circ T_{L-1} \circ \sigma_{L-1} \circ \cdots \sigma_1 \circ T_1 \circ \mathbf{x}, \tag{2.1}$$

where $T_{\mathrm{i}}(\boldsymbol{u}) = W_{\mathrm{i}}\boldsymbol{u} + \boldsymbol{b}_{\mathrm{i}}$, $\mathrm{i} \in [L]$, are affine transformations with unknown parameters $(W_{\mathrm{i}}, \boldsymbol{b}_{\mathrm{i}})$ with $W_{\mathrm{i}} \in \mathbb{R}^{p_{\mathrm{i}} \times p_{\mathrm{i}-1}}$ and $\boldsymbol{b}_{\mathrm{i}} \in \mathbb{R}^{p_{\mathrm{i}}}$, and $\sigma_{\mathrm{i}}(\cdot)$, $\mathrm{i} \in [L]$, are the element-wise activation functions. For simplicity, we use $\mathcal{W}_L = ((W_1, \boldsymbol{b_1}), \ldots, (W_L, \boldsymbol{b_L}))$ to denote all the weight matrices and bias terms. In the above, we denote $L$ as the depth and $p$ as the width of the neural network. Typical examples of activation functions include sigmoid, tanh and Relu. Note that Relu owns good properties, satisfying $\sigma(0) = 0$ and 1-Lipschitz continuousness.

We now introduce the loss function for the purpose of solving and training our model. Feed-forward neural networks can be used to deal with both regression and classification problem. The mainly used loss functions include the quadratic loss for regression and the cross-entropy loss for classification. We minimize the loss function and apply backpropagation to obtain the optimal $\mathcal{W}_L$. Although this minimization problem is often highly non-convex, stochastic gradient descent and its variations, the primary algorithms of training neural networks, can nearly achieve the optimal solution. To accelerate the training process and avoid overfitting, explicit or implicit regularizations on weight matrices have been applied in practice such as weight decay [32], dropout [33], [34], and early stopping [35]. This encourages us to study the capacity bound for DNNs through regularizations on weight matrices.

## 2.2 Generative Models

Among these deep generative models, three major families stand out and deserve a special attention: Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs) and Normalizing Flows.

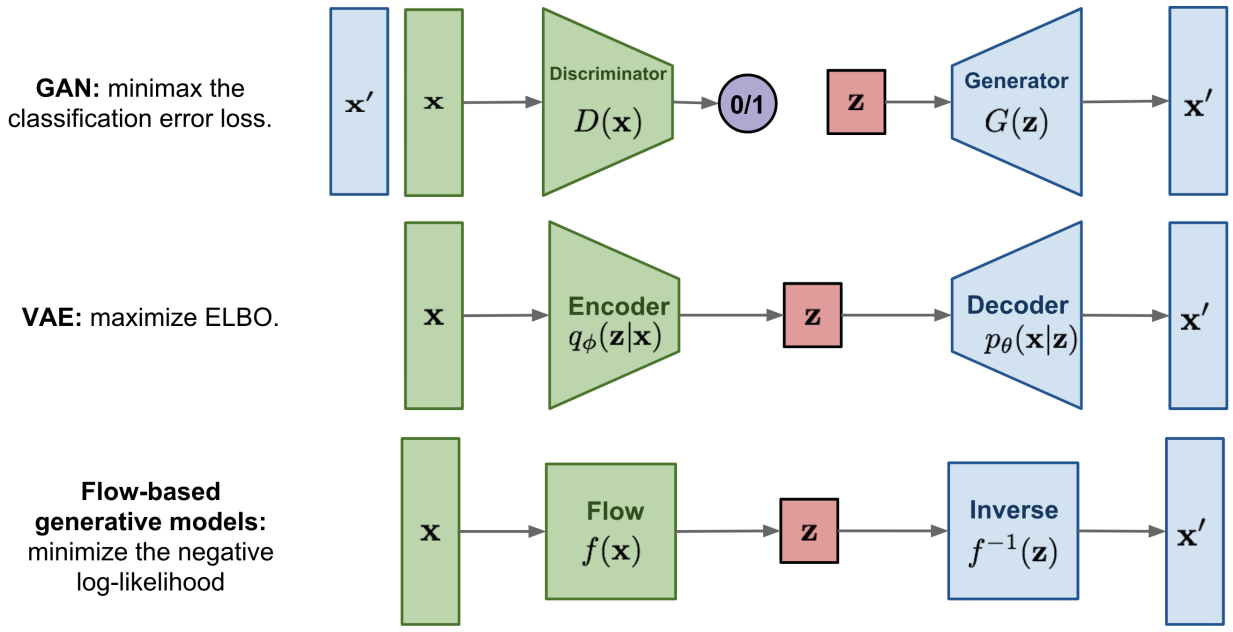



**Figure 2.2.** Three categories of generative models.

### 2.2.1 Generative Adversarial Networks

GANs are the most popular type of neural network architecture that allow neural networks to generate data. They are able to learn a generative model by framing the problem as a supervised learning problem with two sub-models: the generator model that we train to generate new examples, and the discriminator model that tries to classify examples as either real (from the target) or fake (generated). The first plot of Figure 2.2 shows the architecture of GANs. Training GANs is like a two player game. The two networks, the generator and discriminator, are simultaneously learned by pushing the powerful discriminator to distinguish between real data and generative samples. As a result, the generator is trying to maximize its probability of having its outputs recognized as real, while the discriminator is trying to minimize this same value. This leads to the following minimax objective function,

$$\min_{G\in\mathcal{G}} \max_{f\in\mathcal{F}} \mathbb{E}_X[\log(f(X)] + \mathbb{E}_Z[\log(1 - f(G(Z)))], \tag{2.2}$$

where $f \in \mathcal{F}$ is a discriminator, $G \in \mathcal{G}$ is a generator. Optimizing Equation 2.2 is equivalent to minimizing Jensen-Shannon divergence between the generative and data distribution. GANs can generate visually realistic images, but suffer from unstable training and mode collapsing.

The Wasserstein GAN (WGAN) [23] is an extension to the vanilla GAN that improves the stability of training by leveraging the 1-Wasserstein distance between two probability measures. The 1-Wasserstein distance between $P_X$ and $P_{G(Z)}$ is defined as

$$W_1(P_X, P_{G(Z)}) = \inf_{\pi \in \Pi(P_X, P_Z)} \mathbb{E}_{(X,Z)\sim\pi} \|X - G(Z)\|, \tag{2.3}$$

where $\|\cdot\|$ represents the $\ell_2$-norm and $\Pi(P_X, P_Z)$ is the set of all joint distributions of $(X, Z)$ with marginal measures $P_X$ and $P_Z$. It is hard to find the optimal coupling $\boldsymbol{\pi}$ through this constrained primal problem. However, thanks to the Kantorovich-Rubinstein duality, the WGAN can learn the generator $G$ by minimizing the nice dual format

$$W_1(P_X, P_{G(Z)}) = \sup_{f \in \mathcal{F}} \Big\{ \mathbb{E}_X f(X) - \mathbb{E}_Z f(G(Z)) \Big\}, \tag{2.4}$$

where $\mathcal{F}$ is a set of all bounded 1-Lipschitz functions. Weight clipping [23] and gradient penalty [36] are two strategies to maintain the Lipschitz continuity of $f$. Weight clipping utilizes a tuning parameter $c$ to clamp each weight parameter to a fixed interval $[-c, c]$ after each gradient update, but this method is very sensitive to the choice of parameter $c$. Instead, gradient penalty adds a term $\mathbb{E}_X(\|f(X)\| - 1)^2$ in the loss function to enforce the 1-Lipschitz.

### 2.2.2 Variational Auto-Encoders

The VAE [21] defines a "probabilistic decoder" $p_\theta(\mathbf{x}|\mathbf{z})$ with the unknown parameter $\theta$. Then the marginal distribution of $X$ is $p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})dz$. Due to the intractability of this integration, the maximum likelihood estimation is prohibited. Instead, a "probabilistic encoder" $q_\phi(\mathbf{z}|\mathbf{x})$ with the unknown parameter $\phi$ is defined to approximate the posterior

distribution $p_\theta(\mathbf{z}|\mathbf{x}) = p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})/p_\theta(\mathbf{x})$. The objective is to maximize the lower bound of the log likelihood $\log p_\theta(\mathbf{x})$, which is called the *evidence lower bound (ELBO)*:

$$\text{ELBO} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\Big[\log p_\theta(\mathbf{x}|\mathbf{z})\Big] - \text{KL}\Big(q_\phi(\mathbf{z}|\mathbf{x}) \;||\; p(\mathbf{z})\Big),$$

where the first term can be efficiently obtained by the sampling technique and the second term has a closed form expression when $q_\phi$ is Gaussian. VAEs have strong theoretical justifications and typically can cover all modes of the data distribution. However, they often produce blurry images due to the normal approximation of the true poseterior.

The Wasserstein Auto-encoder (WAE) [37] makes two modifications based on the VAE. It uses a deterministic encoder $Q : \mathcal{X} \to \mathcal{Z}$ to approximate the conditional distribution of $Z$ given $X$, and a deterministic encoder $G : \mathcal{Z} \to \mathcal{X}$ to approximate the conditional distribution of $X$ given $Z$. In addition, the WAE adopts the 1-Wasserstein distance between real data $P_X$ and generative distribution $P_{G(Z)}$, rather than the Kullback-Leibler divergence used in VAEs, to train the model. Specifically, it minimizes the following reconstruction error with respect to the generator $G$,

$$\inf_{Q \in \mathcal{Q}} \mathbb{E}_X \|X - G(Q(X))\| + \lambda \mathcal{D}(P_{Q(X)}, P_Z), \tag{2.5}$$

where $\mathcal{D}$ is any divergence measure between two distributions $P_{Q(X)}$ and $P_Z$, and $\lambda > 0$ is a regularization coefficient. The regularization term distributionally forces the aggregated posterior $P_{Q(X)}$ to match the prior distribution $P_Z$.

### 2.2.3 Normalizing Flows

A Normalizing Flow is a transformation of a simple probability distribution, such as a standard normal, into a more complex distribution by a sequence of invertible and differentiable mappings. The density of a sample can be evaluated by transforming it back to the original simple distribution and then computing the product of i) the density of the inverse-transformed sample under this distribution and ii)the associated change in volume induced by the sequence of inverse transformations.

In detail, Let $Z \in \mathbb{R}^d$ be a random variable with a known and tractable probability density function $P_Z : \mathbb{R}^d \to \mathbb{R}$. Let $G : \mathbb{R}^d \to \mathbb{R}^d$ be an invertible function and $X = G(Z)$. Then using the change of variables formula, one can compute the probability density function of the random variable $X$:

$$P_X(\mathbf{x}) = P_Z(Q(\mathbf{x}))|det(\mathrm{D}Q(\mathbf{x}))| = P_Z(Q(\mathbf{x}))|det(\mathrm{D}G(Q(X)))|^{-1}$$

where $Q$ is the inverse of $G$, $\mathrm{D}Q(\mathbf{x}) = \partial Q/\partial \mathbf{x}$ is the Jacobian of $Q$ and $\mathrm{D}G(\mathbf{z}) = \partial G/\partial \mathbf{z}$ is the Jacobian of $G$. Maximizing the log likelihood of $P_X(\mathbf{x})$ can obtain the model $Q$. However, constructing arbitrarily complicated non-linear invertible functions (bijections) can be difficult. One approach to this is to note that the composition of invertible functions is itself invertible and the determinant of its Jacobian has a specific form. Let $G_1, \ldots, G_N$ be a set of $N$ bijective functions and define $G = G_N \circ G_{N-1} \circ \cdots \circ G_1$ to be the composition of the functions. Then it can be shown that $G$ is also bijective, with inverse

$$Q = Q_1 \circ \cdots \circ Q_{N-1} \circ Q_N$$

and the determinant of the Jacobian is

$$\det(\mathrm{D}Q(\mathbf{x})) = \prod_{\mathrm{i}=1}^{N} \det(\mathrm{D}Q_{\mathrm{i}}(\mathbf{x}_{\mathrm{i}})),$$

where $\mathbf{x}_{\mathrm{i}}$ denotes the value of the i-th intermediate flow as $\mathbf{x}_{\mathrm{i}} = G_{\mathrm{i}} \circ \cdots \circ G_1(\mathbf{z}) = Q_{\mathrm{i}+1} \circ \cdots \circ Q_N(\mathbf{x})$ and so $\mathbf{x}_N = X$. Thus, a set of nonlinear bijective functions can be composed to construct successively more complicated functions. *Affine coupling layers* [25], [38], [39] are one of the most popular methods such that the Normalizing Flow is sufficiently expressive to model the distribution of interest, and computationally efficient, both in terms of computing $Q$ and $G$ but also in terms of the calculation of the determinant of the Jacobian.

# 3. STATISTICAL LEARNING

## 3.1 Overview

In supervised learning problems such as classification and regression, our target is to predict an output $y \in \mathcal{Y}$ based on a set of features $\mathbf{x} \in \mathcal{X}$. Informally, we choose a predictor $f : \mathcal{X} \to \mathcal{Y}$ from the hypothesis class $\mathcal{F}$ such that $f(\mathbf{x})$ is a good prediction of $y$. Let $(\mathbf{x}, y)$ be from an unknown distribution $P$, and the loss function be $g(\cdot, \cdot) : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$. Define the *expected risk* as

$$R(f) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \left[ g(f(\mathbf{x}), y) \right],$$

and our goal is to find the expected risk minimizer which is denoted by $f^* \in \operatorname{argmin}_{f\in\mathcal{F}} R(f)$. Given $m$ i.i.d. samples $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$, where each pair is from $P$ over $\mathcal{X} \times \mathcal{Y}$, the approximation of $f^*$ is obtained by minimizing the *empirical risk*:

$$\hat{R}(f) = \frac{1}{m} \sum_{\mathrm{i}=1}^{m} g(f(\mathbf{x}_\mathrm{i}), y_\mathrm{i}). \tag{3.1}$$

The trained predictor $f$ is also called the *empirical risk minimizer* (ERM) defined as any hypothesis $f \in \mathcal{F}$ that minimizes Equation 3.1 $\hat{f} \in \operatorname{argmin}_{f\in\mathcal{F}} \hat{R}(f)$. In practice, we often choose the quadratic loss for regression problems, while the hinge loss and the cross entropy loss are commonly used in classification tasks. Statistical learning is an active area of research in the past two decades: well-known monographs in this area include [40]–[42].

The key question of statistical learning is to analyze and control the *excess risk*, which is the difference between $R(f^*)$ and $R(\hat{f})$. The excess risk characterizes the gap between the expected risk of $\hat{f}$ and the optimal $f^*$. Another related concept is called the *generalization error*, which is the difference between $R(\hat{f})$ and $\hat{R}(\hat{f})$. Mathematically, the generalization error is a measure of how accurately an algorithm is able to predict outcome values for previously unseen data. The generalization error can be minimized by avoiding *overfitting* in the learning algorithm. We will show later that the generalization error is easy to control if the excess risk is bounded. So, how do we analyze the excess risk? Note that the excess risk is a random variable depending on the training set via $\hat{f}$, and the sample size $m$ is finite. Therefore, the central limit theorem in asymptotics cannot be directly applied here. We

formulate the analysis as a probability statement. Given $\theta \in (0,1)$, the excess risk is upper bounded by some $\epsilon$ with probability at least $1-\theta$, thaft is, $\mathbb{P}\left(R(\hat{f}) - R(f^*) \leq \epsilon\right) \geq 1-\theta$, or equivalently,

$$\mathbb{P}\left(R(\hat{f}) - R(f^*) \geq \epsilon\right) \leq \theta, \tag{3.2}$$

where $\epsilon$ is a function relying on $\theta$ and the complexity of the hypothesis class $\mathcal{F}$.

To explicitly describe $\epsilon$, we rewrite the excess risk as

$$R(\hat{f}) - R\left(f^*\right) = \underbrace{R(\hat{f}) - \hat{R}(\hat{f})}_{(a)} + \underbrace{\hat{R}(\hat{f}) - \hat{R}\left(f^*\right)}_{(b)} + \underbrace{\hat{R}\left(f^*\right) - R\left(f^*\right)}_{(c)}.$$

Term (b), $\hat{R}(\hat{f}) - \hat{R}(f^*)$, is non-positive, because $\hat{f}$ is chosen to minimize $\hat{R}(f)$ in the hypothesis class $\mathcal{F}$. Term (c) is the difference between a sample average and an expectation in terms of the fixed function $f^*$, such that

$$\hat{R}\left(f^*\right) - R\left(f^*\right) = \frac{1}{m}\sum_{\mathrm{i}=1}^{m} g\left(f^*(\mathbf{x}_\mathrm{i}), y_\mathrm{i}\right) - \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[g\left(f^*(\mathbf{x}), y\right)\right].$$

The law of large numbers shows that this term converges to zero. With information about the tails of $g(f^*(\mathbf{x}), y)$ such as boundedness, we can use concentration inequalities to bound its value. Term (a), $R(\hat{f}) - \hat{R}(\hat{f})$, is more interesting and complicated, since $\hat{f}$ is random based on the chosen data. An easy approach is to provide a uniform upper bound,

$$R(\hat{f}) - \hat{R}(\hat{f}) \leq \sup_{f\in\mathcal{F}}\left|R(f) - \hat{R}(f)\right|,$$

which motivates us to study the *uniform convergence*. Suppose we can ensure that $R(f)$ and $\hat{R}(f)$ were close (say within $\epsilon/2$ ) for all $f \in \mathcal{F}$. Then, we could guarantee that $R(\hat{f})$ and $\hat{R}(\hat{f})$ were within $\epsilon/2$, as well as $R(f^*)$ and $\hat{R}(f^*)$. Therefore, Equation 3.2 can be written formally as:

$$\mathbb{P}\left(R(\hat{f}) - R\left(f^*\right) \geq \epsilon\right) \leq \mathbb{P}\left(\sup_{f\in\mathcal{F}}\left|R(f) - \hat{R}(f)\right| \geq \frac{\epsilon}{2}\right). \tag{3.3}$$

On the right-hand side is a statement about uniform convergence, which describes the probability of the event that the largest difference between the empirical and expected risk is at

least $\epsilon/2$ , or equivalently, the event that this difference exceeds $\epsilon/2$ for at least one $f \in \mathcal{F}$. Using uniform convergence, we bound the difference between the test error and training error of any $f$ by the complexity of $\mathcal{F}$, that is,

$$|R(f) - \hat{R}(f)| \leq O_p\left(\sqrt{\frac{\text{Complexity } (\mathcal{F})}{m}}\right)$$

To describe the complexity of function class $\mathcal{F}$, several tools have been developed such as VC dimension, covering number and Rademacher complexity that we will cover later.

### 3.2 Uniform Convergence

In this section, we will introduce some commonly used techniques for establishing uniform convergence. One of the most important tools is the McDiarmid's inequality, which is used to bound not the average of random variables $X_1 \ldots, X_m$, but any function on $X_1, \ldots, X_m$ satisfying an appropriate bounded differences condition.

**Theorem 3.2.1** (McDiarmid's inequality)**.** *Let $X_1, \ldots, X_m$ be independent random variables with support on $\mathcal{X}$. Let $f : \mathcal{X}^m \mapsto \mathbb{R}$ be a function satisfying the following bounded difference condition,*

$$(\forall \mathrm{i}, \forall \mathbf{x}_1, \ldots, \mathbf{x}_m, \mathbf{x}_\mathrm{i}') \quad |f(\mathbf{x}_1, \ldots, \mathbf{x}_\mathrm{i}, \ldots, \mathbf{x}_m) - f(\mathbf{x}_1, \ldots, \mathbf{x}_\mathrm{i}', \ldots, \mathbf{x}_m)| \leq B_\mathrm{i},$$

*then,*

$$\mathbb{P}\left(|f(X_1, \ldots, X_m) - \mathbb{E}[f(X_1, \ldots, X_m)]| \geq t\right) \leq 2\exp\left(-\frac{2t^2}{\sum_\mathrm{i} B_\mathrm{i}^2}\right).$$

We apply martingale to prove this inequality, and please refer to [40] for more details. This is a quite powerful result, as it holds for any independent random variables, even if $f$ is complex such as neural networks. As long as the function is not too sensitive to perturbations in one of its arguments, we get good concentration.

To have a better understanding, we now use them to analyze the generalization results for a finite hypothesis class. This is accomplished by a two-step concentration and the union bound.

**Example 3.2.1.** *Let $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$ be $m$ i.i.d samples from the unknown distribution $\mathcal{D}$. Assume that $\mathcal{F}$ is a finite hypothesis class, i.e. $\mathcal{F} = \{f_1, \ldots, f_k\}$ where $k < \infty$ and $f_{\mathrm{j}} : \mathcal{X} \mapsto \mathcal{Y}$ for $\forall \mathrm{j}$. Let $g$ be the zero-one loss, i.e. $g(f(\mathbf{x}), y) = \mathbb{I}[f(\mathbf{x}) \neq y]$. Let $\hat{f}$ be the empirical risk minimizer. For fix $\theta \in (0, 1)$, with probability at least $1 - \theta$, we have*

$$R(\hat{f}) - R\left(f^*\right) \leq \sqrt{\frac{2(\log k + \log(2/\theta))}{m}}.$$

### 3.2.1 Rademacher Complexity

In the previous section, we analyzed the excess risk with a finite hypothesis class $\mathcal{F}$, i.e., $|\mathcal{F}| < \infty$. However, the union bound cannot be applied to infinite hypothesis classes. This motivates us to explore more sophisticated approaches to measure the capacity of a hypothesis class. It leads to a introduction to a framework called *Rademacher complexity* to uniformly bound the difference between the expected and empirical risk for any $f \in \mathcal{F}$.

**Definition 3.2.1.** *The **empirical** Rademacher complexity of the function class $\mathcal{F}$ with respect to a data set $\{\mathbf{x}_1 \ldots \mathbf{x}_m\}$ is defined as:*

$$\widehat{\mathfrak{R}}_m(\mathcal{F}) = \mathbb{E}_\delta \left[ \sup_{f \in \mathcal{F}} \left( \frac{1}{m} \sum_{\mathrm{i}=1}^{m} \delta_{\mathrm{i}} f(\mathbf{x}_{\mathrm{i}}) \right) \right],$$

*and the Rademacher complexity is defined as:*

$$\mathfrak{R}_m(\mathcal{F}) = \mathbb{E}_{\delta, \mathbf{x}} \left[ \sup_{f \in \mathcal{F}} \left( \frac{1}{m} \sum_{\mathrm{i}=1}^{m} \delta_{\mathrm{i}} f(\mathbf{x}_{\mathrm{i}}) \right), \right]$$

*where $\delta = \{\delta_1, \ldots, \delta_m\}$ are $m$ independent Rademacher random variables, that is $\mathbb{P}[\delta_{\mathrm{i}} = -1] = \mathbb{P}[\delta_{\mathrm{i}} = +1] = 1/2$.*

Here we give an intuitive explanation about the Rademacher complexity. Consider the simple binary classification problem with inputs $\mathbf{x}_1, \ldots, \mathbf{x}_m$. If the corresponding labels are random $\delta_1, \ldots, \delta_m$, this becomes a meaningless learning problem. Therefore, the Rademacher complexity is used to capture how well the best function from the function class $\mathcal{F}$ can fit

these random labels. A large $\mathcal{F}$ will be able to fit noise better and thus have a larger Rademacher complexity. In practice, we would like $\mathfrak{R}_m(\mathcal{F})$ to go to zero as $m$ increases.

The basic properties of Rademacher complexity are listed as follows.

- $\mathfrak{R}_m(\mathcal{F}) = 0$ for $\mathcal{F} = \{f\}$.
- $\mathfrak{R}_m(\mathcal{F}_1) \leq \mathfrak{R}_m(\mathcal{F}_2)$ if $\mathcal{F}_1 \subseteq \mathcal{F}_2$.
- $\mathfrak{R}_m(\mathcal{F}_1 + \mathcal{F}_2) \leq \mathfrak{R}_m(\mathcal{F}_1) + \mathfrak{R}_m(\mathcal{F}_2)$ for $\mathcal{F}_1 + \mathcal{F}_2 = \{f_1 + f_2 : f_1 \in \mathcal{F}_1, f_2 \in \mathcal{F}_2\}$
- $\mathfrak{R}_m(c\mathcal{F}) = |c|\mathfrak{R}_m(\mathcal{F})$

Next, we show the crucial theorem that links the uniform convergence and Rademacher complexity.

**Theorem 3.2.2.** *Let $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$ be $m$ i.i.d. samples drawn from the unknown distribution $\mathcal{D}$. Let $\mathcal{F}$ be a hypothesis class and $g$ be the loss function where $g \circ \mathcal{F}$ belongs to $\{g(f(\mathbf{x}), y) \mid g \circ f : \mathcal{X} \times \mathcal{Y} \to [0, 1], f \in \mathcal{F}\}$. Fix $\theta \in (0, 1)$. With probability at least $1 - \theta$, we have:*

$$(\forall f \in \mathcal{F})\ R(f) \leq \widehat{R}(f) + 2\mathfrak{R}_m(g \circ \mathcal{F}) + \sqrt{\frac{\log(1/\theta)}{2m}}.$$

**Theorem 3.2.3** (Ledoux-Talagrand contraction inequality)**.** *Assume that the function class $\mathcal{F} \subseteq \{f \mid f : \mathcal{X} \to \mathbb{R}\}$. Assume that the function $\phi : \mathbb{R} \to \mathbb{R}$ is $M$-Lipschitz continuous. Define the shorthand notation: $\phi(\mathcal{F}) = \{\phi(f) \mid f \in \mathcal{F}\}$. We have:*

$$\widehat{\mathfrak{R}}_m(\phi(\mathcal{F})) \leq M\widehat{\mathfrak{R}}_m(\mathcal{F})$$

The Ledoux-Talagrand contraction inequality is quite useful when analyzing the Rademacher complexity of loss class $g \circ \mathcal{F}$, since we can transfer it to analyze the complexity of our hypothesis class $\mathcal{F}$. Equipped with above theorems, we can bound Equation 3.2 with probability at least $1 - \theta$

$$R(\hat{f}) - R(f^*) \leq 4\mathfrak{R}_m(g \circ \mathcal{F}) + \sqrt{\frac{2\log(2/\theta)}{m}}.$$

Once we have the Rademacher complexity of hypothesis class $\mathcal{F}$, we can easily bound the difference between empirical risk and expected risk as well as the excess risk. There are several tools helping control $\mathfrak{R}_m(\mathcal{F})$ such as VC dimension and covering number introduced in the next section.

### 3.2.2 Growth function and VC dimension

So far, we have set up Rademacher complexity as a measure of the capacity of infinite hypothesis class. Let us instantiate Rademacher complexity when the function class has finite possible outputs such as binary classification problem. Assume the dataset $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ contains $m$ i.i.d. samples from distribution $P_X$. In general, we assume a function class $\mathcal{F} \subseteq \{f | f : \mathcal{X} \to \{0, 1\}\}$. We introduce the following shorthand notation: $\mathcal{F}(S) = \{(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_m)) \in \{0, 1\}^m | f \in \mathcal{F}\}$. That is, $\mathcal{F}(S)$ contains all the $\{0, 1\}^m$ vectors that can produced by applying all functions in $\mathcal{F}$ to the dataset $S$.

**Definition 3.2.2** (Growth Function)**.** *The growth function (or shatter coefficient) of a class of functions $\mathcal{F} \subseteq \{f | f : \mathcal{X} \to \{0, 1\}\}$ for $m$ samples is:*

$$G(\mathcal{F}, m) = \max_{S \in \mathcal{X}^m} |\mathcal{F}(S)|.$$

For boolean functions, if $G(\mathcal{F}, m) = 2^m$, meaning we obtain all possible labels, we say $\mathcal{F}$ shatters any $m$ points $z_1, \ldots, z_m$ that achieve the maximum of $\mathcal{F}(S)$. One advantage of growth function is that it turns the infinite function class to a finite coefficient. Therefore, we can directly use the following Massart's finite lemma to link with Rademacher complexity.

**Lemma 3.2.1** (Massart's Finite Lemma)**.** *For $\mathcal{A} \subseteq \mathbb{R}^m$ with $R^2 = \frac{\max_{\boldsymbol{a} \in \mathcal{A}} \|\boldsymbol{a}\|_2^2}{m}$,*

$$\mathbb{E}\left[\sup_{\boldsymbol{a} \in \mathcal{A}} \frac{1}{m} \sum_{\mathrm{i}=1}^{m} \delta_{\mathrm{i}} a_{\mathrm{i}}\right] \leq \sqrt{\frac{2R^2 \log |\mathcal{A}|}{m}},$$

*where $\delta = \{\delta_1, \ldots, \delta_m\}$ are $m$ i.i.d Rademacher random variables.*

Taking $\mathcal{A} = \mathcal{F}(S)$, we have $R^2 \leq 1$. By Massart's finite lemma, it is straightforward that

$$\widehat{\mathfrak{R}}_m(\mathcal{F}) \leq \sqrt{\frac{2 \log G(\mathcal{F}, m)}{m}}.$$

Thus, to get meaningful bounds, we want $G(\mathcal{F}, m)$ to grow sub-exponentially with $m$. Otherwise, the Rademacher complexity will not go to zero, and we will not obtain uniform convergence. This is expected since if $\mathcal{F}$ can really hit all labels for all $m$, we would be able to fit any label of the data, leading to massive overfitting.

Although the growth function nicely captures the behavior of an infinite $\mathcal{F}$, it is not necessarily the most convenient quantity to get a handle on. In the following, we use a concept called *VC dimension* to gain more intuition about the growth function.

**Definition 3.2.3.** *The VC dimension of a class of functions $\mathcal{F}$ with Boolean outputs is the maximum number of points that can be shattered by $\mathcal{F}$:*

$$VC(\mathcal{F}) = \max_{m \in \mathbb{M}} \{m | G(\mathcal{F}, m) = 2^m\}$$

**Lemma 3.2.2** (Sauer-Shelah Lemma)**.** *For a function class $\mathcal{F}$ with Boolean outputs and VC dimension $d$, then we have*

$$G(\mathcal{F}, m) \leq \sum_{\mathrm{i}=0}^{d} \begin{pmatrix} m \\ \mathrm{i} \end{pmatrix} \leq (m+1)^d.$$

Combining this theorem with the previous conclusions, we have

$$\widehat{\mathfrak{R}}_m(\mathcal{F}) \leq \sqrt{\frac{2 \log G(\mathcal{F}, m)}{m}} \leq \sqrt{\frac{2VC(\mathcal{F}) \log(m+1)}{m}}.$$

### 3.2.3 Covering Number

For infinite hypothesis classes, we observe that growth function and VC dimension are appropriate measures since all that mattered was the behavior of a function class on a finite set of points. However, these two approaches only work for functions that return a finite

number of values. Can we retain the combinatorial nature of growth function, but allow for real-valued functions such as regression problems? We explore covering numbers in the section to solve this problem. Covering numbers count the number of balls of size $\epsilon$ one needs to cover the hypothesis class, then the Massart's finite lemma can be applied to control the representatives. In essence, covering numbers allow us to discretize the problem.

**Definition 3.2.4.** *A metric space $(\mathcal{Y}, \rho)$ is a set $\mathcal{Y}$ and a function $\rho : \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$ satisfying*

- *Identity of indiscernibles:* $\rho(y, y) = 0$
- *Symmetry:* $\rho(y, z) = \rho(z, y)$
- *Triangle inequality:* $\rho(y, z) \leq \rho(y, x) + \rho(x, z)$

If $\rho(x, y) = 0$ is possible for $x \neq y$, then we say $\rho$ is a pseudometric. In this section, we will work with the pseudometric. For example, the pseudometric for a set of functions $\mathcal{F}$ mapping from $\mathcal{X}$ to $\mathbb{R}$ is $\rho_m(f, f') = \|f - f'\|_{L_2(P_m)} := (\frac{1}{m} \sum_{\mathrm{i}=1}^{m} (f(\mathbf{x}_\mathrm{i}) - f'(\mathbf{x}_\mathrm{i}))^2)^{1/2}$.

**Definition 3.2.5.** *The $\tau$-covering number $N(\mathcal{F}, \tau, \rho)$ of a class of function $\mathcal{F} \subseteq \mathcal{A}$ with respect to the metric $\rho$ is the size of the smallest cover:*

$$\min\{n : \exists\{f_1, \ldots, f_n\} \subseteq \mathcal{A}, \mathcal{F} \subseteq \cup_{\mathrm{i}=1}^{n} \mathcal{B}_\tau^\rho(f_\mathrm{i})\},$$

*where $\mathcal{B}_\tau^\rho(f_\mathrm{i})$ is the ball with radius $\tau > 0$ centered at $f_\mathrm{i} \in \mathcal{A}$, defined as $\mathcal{B}_\tau^\rho(f_\mathrm{i}) = \{f' \in \mathcal{A} : \rho(f_\mathrm{i}, f') \leq \tau\}$.*

For the metric $\rho$, if $\mathcal{A}$ is a family of functions mapping $\mathbb{R}^p$ to $\mathbb{R}$, we define the metric $\rho_m = (\frac{1}{m} \sum_{\mathrm{i}=1}^{m} (f(\mathbf{x}_\mathrm{i}) - f'(\mathbf{x}_\mathrm{i}))^2)^{\frac{1}{2}}$, or the metric $\rho_\infty = \sup_{\mathrm{i} \in [m]} |f(\mathbf{x}_\mathrm{i}) - f'(\mathbf{x}_\mathrm{i})|$, where $\mathbf{x} \in \mathbb{R}^p$. When $\mathcal{A}$ maps a class of functions from $\mathbb{R}^p$ to $\mathbb{R}^r$, we generalize the definition as $\rho_m = (\frac{1}{m} \sum_{\mathrm{i}=1}^{m} \left\| f(\mathbf{x}_\mathrm{i}) - f'(\mathbf{x}_\mathrm{i}) \right\|_2^2)^{\frac{1}{2}}$, or $\rho_\infty = \sup_{\mathrm{i} \in [m]} \left\| f(\mathbf{x}_\mathrm{i}) - f'(\mathbf{x}_\mathrm{i}) \right\|_2$.

From above definitions, it is straightforward that as $\tau$ decreases, $f'$ in the cover $\mathcal{F}'$ is a better approximation of $f$, but $N(\mathcal{F}, \tau, \rho)$ also increases. In general, we would like $N(\mathcal{F}, \tau, \rho)$ to be small, so what is the trade-off? The following theorems establish that the covering

number enable to upper bound the Rademacher complexity, which also provide hints for this trade-off.

**Theorem 3.2.4** (Dudley's theorem)**.** *Let $\mathcal{F}$ be a family of functions $f : \mathcal{X} \to \mathbb{R}$. Consider the $\rho_m$ pseudometric on $\mathcal{F}$, then*

$$\widehat{\mathfrak{R}}_m(\mathcal{F}) \leq 12 \int_0^\infty \sqrt{\frac{2 \log N(\mathcal{F}, \tau, \rho_m)}{m}} d\epsilon.$$

The theorems we mentioned above are the main techniques that we use to obtain the generalization bound for neural networks, which has been extensively studied in literature, especially norm-constrained fully connected DNNs [43]–[49]. In particular, spectral norm-constrained fully connected DNNs were studied in [45], [47], [49]. Assume that the spectral norm of the weight matrix in each layer equals to 1, and the width of each hidden layer is $p$. Then the corresponding bound of generalization error is of order $\sqrt{p^3 L^2/m}$ [45], [47] and $\sqrt{pLr/m}$ [49], respectively, where $m$ is the sample size, $L$ is the depth, and $r$ is the rank of weight matrices. On the other hand, a lower bounds for the generalization error with an order of $\sqrt{p/m}$ is established in [48]. In addition, some special cases of the matrix mixed $L_{p,q}$ norm-constrained fully connected DNNs were studied in [43], [44], [46], [48], [50]. For example, [44] provided an exponential bound on the width $p$ based on the Frobenius norm of the weight matrices; [47] provided a polynomial bound on $L$ and $p$ based on the spectral norm and the $L_{2,1}$ norm.

# 4. GENERALIZATION ERROR BOUNDS ON ADVERSARIAL LEARNING OF DEEP NEURAL NETWORKS

As we mentioned in chapter 1, the generalization behavior under the adversarial attacks are much worse than the situation without adversarial perturbations. In this work, we study the adversarial robust generalization property of DNNs to theoretically gain deeper understanding of this problem.

We concentrate on $\ell_\infty$ adversarial attacks produced under the Fast Gradien Sign Method (FGSM) [16], which proposes to compute the adversarial examples via the gradient of the loss on clean data $\mathbf{x}$, i.e.,

$$\tilde{\mathbf{x}}_{\mathrm{i}} = \mathbf{x}_{\mathrm{i}} + \epsilon \cdot \mathrm{sign}\Big(\nabla_{\mathbf{x}} g(f(\mathbf{x}_{\mathrm{i}}), y_{\mathrm{i}})\Big), \text{ for } \mathrm{i} = 1, \ldots, m. \tag{4.1}$$

Our goal is to study the generalization behavior of the FGSM, i.e., the difference between the expected adversarial risk and the empirical adversarial risk for any $f \in \mathcal{F}$

$$\mathbb{E}_{(\mathbf{x},y)}[g(f(\tilde{\mathbf{x}}), y)] - \frac{1}{m}\sum_{\mathrm{i}=1}^{m} g(f(\tilde{\mathbf{x}}_{\mathrm{i}}), y_{\mathrm{i}}). \tag{4.2}$$

Specifically, we provide tight sample complexity bounds for adversarially robust generalization of DNNs based on both spectral norms and ranks of weight matrices under the $\ell_\infty$ adversarial attacks. Our main contributions can be summarized as follows.

1. By novelly viewing a DNN as a composition of a shallow network and a Lipschitz continuous function on a low dimension, we achieve a tighter upper bound on the Rademacher complexity of the DNN class with spectral normalization and low-rank weight matrices under the FGSM attack. This bound is depth-free comparing to existing works [19], [20] that polynomially depend on the depth.

2. We compare the adversarial Rademacher complexity with the natural Rademacher complexity. The adversarial complexity is never smaller than its natural counterpart, but the effect of adversarial perturbations can be limited under our weight normalization framework.

3. We conduct experiments on neural networks with different depth to verify our theoretical findings. In particular, our numerical results establish that the adversarial generalization bound is depth-free if there exists a low-rank weight matrix, and the adversarial generalization error is proportional to the attack size $\epsilon$.

## 4.1 Related Works

### 4.1.1 The Spectral Norm and The Rank in DNN

For $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^{p_0}$, let $f_{\mathcal{W}_L}(\mathbf{x})$ denote a DNN of the form

$$f_{\mathcal{W}_L}(\mathbf{x}) = W_L \sigma_{L-1}(W_{L-1} \cdots \sigma_1(W_1 \mathbf{x})),$$

where $\mathcal{W}_L = (W_1, \ldots, W_L)$ are weight matrices, and $\sigma_{\mathrm{i}}(\cdot)$ are element-wise nonlinear activation functions, satisfying $\sigma_{\mathrm{i}}(\mathbf{0}) = \mathbf{0}$, for $\mathrm{i} = 1, \ldots, L$. Here $L$ denotes the depth of the neural network, and $p$ is the width that is the maximal row or column dimension of $W_1, \ldots, W_L$. Notice that the spectral norms of weight matrices of a DNN reflect the Lipschitz coefficient of $f_{\mathcal{W}_L}$. The spectral weight normalization for DNNs has achieved remarkable successes in many complex learning tasks [51], [52]. The advantages of using spectral normalization include that Lipschitz constant is the only hyper-parameter to be tuned and implementation is simple with a small additional computational burden. Moreover, many implicit regularization methods have been used for training DNNs such as dropout [33] and early stopping [35]. Recently, [53], [54] have shown that dropout can be treated as a low-rank regularizer with data dependent singular-value threshold. All these promising results motivate us to consider the class of DNNs with constraints on both spectral norms and ranks of weight matrices. In addition, the following observation based on the compositional structure of DNNs is crucial for our theoretical development. The SVD of $W_l = U_l \Sigma_l V_l^\top$ gives the function

$$f_{\mathcal{W}_L}(\mathbf{x}) = W_L \sigma_{L-1}(\cdots \sigma_l(U \Sigma V^\top \sigma_{l-1}(\cdots \sigma_1(W_1 \mathbf{x})).$$

This function is the composition of a $r_l$-dimensional Lipschitz continuous function

$$W_L\sigma_{L-1}(W_{L-1}\cdots\sigma_l(U\Sigma\mathbf{x})),$$

and a depth $l$ network

$$V^\top\sigma_{l-1}(W_{l-1}\cdots\sigma_1(W_1\mathbf{x})),$$

where $r_l$ is the rank of $W_l$. This decomposition, which is related to both spectral norms and ranks of weight matrices, reveals some intrinsic structures of DNNs.

### 4.1.2 Natural Learning and Adversarial Robust Learning

Consider the class of DNNs with constraints on both spectral norms and ranks of weight matrices,

$$\mathcal{F}_{\mathcal{W}_L} = \Big\{ f_{\mathcal{W}_L} : \big\|W_{\mathrm{j}}\big\|_2 \le c_{\mathrm{j}}, \mathrm{rank}(W_{\mathrm{j}}) \le r_{\mathrm{j}}, \forall \mathrm{j} \in [L] \Big\},$$

where $c_{\mathrm{j}}$'s are the upper bounds on the spectral norms of corresponding weight matrices, and $r_{\mathrm{j}}$'s are the upper bounds on the ranks. Let $g(f_{\mathcal{W}_L}(\cdot),\cdot) : \mathcal{X}\times\mathcal{Y}\to\mathbb{R}$ be the loss function. We first introduce the function class

$$g\circ\mathcal{F}_{\mathcal{W}_L} := \Big\{ (\mathbf{x},y)\mapsto g(f_{\mathcal{W}_L}(\mathbf{x}),y) : f_{\mathcal{W}_L}\in\mathcal{F}_{\mathcal{W}_L} \Big\}.$$

The goal of *natural learning* is to find $f_{\mathcal{W}_L}\in\mathcal{F}_{\mathcal{W}_L}$ such that the population risk

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[g(f_{\mathcal{W}_L}(\mathbf{x}),y)]$$

is minimized. Recall that $\tilde{\mathbf{x}} = \mathbf{x} + \epsilon\cdot\mathrm{sign}(\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x})),y)$ is the adversarial example via the FGSM, so $\tilde{\mathbf{x}}$ is a function of $(\mathbf{x},y)$, $f_{\mathcal{W}_L}$, and $g$. Define the class of DNNs under the FGSM as

$$\widetilde{\mathcal{F}}_{\mathcal{W}_L} = \Big\{ (\mathbf{x},y)\mapsto f_{\mathcal{W}_L}(\tilde{\mathbf{x}}) : f_{\mathcal{W}_L}\in\mathcal{F}_{\mathcal{W}_L},\ \tilde{\mathbf{x}} = \mathbf{x} + \epsilon\cdot\mathrm{sign}(\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}),y)) \Big\}. \tag{4.3}$$

Combining with the loss function, we similarly introduce another function class

$$g \circ \widetilde{\mathcal{F}}_{\mathcal{W}_L} = \Big\{ (\mathbf{x}, y) \mapsto g(f_{\mathcal{W}_L}(\tilde{\mathbf{x}}), y) : f_{\mathcal{W}_L} \in \mathcal{F}_{\mathcal{W}_L}, \tilde{\mathbf{x}} = \mathbf{x} + \epsilon \cdot \operatorname{sign}(\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}), y)) \Big\}.$$

The goal of *adversarial robust learning* is to find $f_{\mathcal{W}_L} \in \mathcal{F}_{\mathcal{W}_L}$ such that the population adversarial risk

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[g(f_{\mathcal{W}_L}(\tilde{\mathbf{x}}), y)]$$

is minimized. In practice, we minimize the adversarial empirical risk to obtain the $f_{\mathcal{W}_L}$. We care about the upper bound of Equation 4.2 so that the adversarial empirical risk can be close to the adversarial population risk.

The key challenge for adversarial learning is that the adversarial example $\tilde{\mathbf{x}}$ depends on both neural network $f_{\mathcal{W}_L}$ and loss function $g$, so deriving the generalization upper bound is more complex. It should be pointed out that our theoretical development can be easily extended to adversarial examples obtained by the $k$-step PGD method.

## 4.2 Generalization Bounds for Adversarial Learning

### 4.2.1 An Upper Bound on Rademacher Complexity for Adversarial Learning

In this section, we establish an upper bound on the Rademacher complexity for the function class $g \circ \widetilde{\mathcal{F}}_{\mathcal{W}_L}$. Key technical tools are covering numbers and the Dudley's entropy integral that we introduced before. We first establish the covering number of the function class $\widetilde{\mathcal{F}}_{\mathcal{W}_L}$.

**Lemma 4.2.1.** *Assume the activation function $\sigma(\cdot)$ is 1-Lipschitz and 1-smooth, and the loss function $g(\cdot, y)$ is 1-Lipschitz and 1-smooth for any fixed label $y$. Let $\mathbf{x}_{\mathrm{i}} \in \mathcal{X} = \{\mathbf{x} \in \mathbb{R}^p : \|\mathbf{x}\| \leq B\}$ for $\mathrm{i} = 1, \ldots, m$. Then the covering number of $\widetilde{\mathcal{F}}_{\mathcal{W}_L}$ with respect to the metric $\rho_m$ satisfies*

$$N(\widetilde{\mathcal{F}}_{\mathcal{W}_L}, \tau, \rho_m) \leq \left( \frac{9L(B + \sqrt{p}\epsilon + \Gamma) \prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}}}{\tau} \right)^{(2p+1)\sum_{\mathrm{j}=1}^{L} r_{\mathrm{j}}},$$

*where*

$$\Gamma = \epsilon \frac{1}{\kappa} \left( 1 + \frac{1}{\kappa} \prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}} \right) \prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}} \left( 1 + \frac{B}{L} \sum_{\mathrm{j}=1}^{L} \left( \prod_{k=1}^{\mathrm{j}} c_k \right) \right),$$

*and* $\kappa \leq \min_{t\in[p]} |\nabla^{(t)}_{\mathbf{x}_\mathrm{i}} g(f_{\mathcal{W}_L}(\mathbf{x}_\mathrm{i}), y_\mathrm{i})|$ *with* $\nabla^{(t)}_{\mathbf{x}_\mathrm{i}}$ *being the* $t$*-th element of* $\nabla_{\mathbf{x}_\mathrm{i}}$ *for* $t = 1, \ldots, p$.

Here we require that $|\nabla^{(t)}_{\mathbf{x}_\mathrm{i}} g(f_{\mathcal{W}_L}(\mathbf{x}_\mathrm{i}), y_\mathrm{i})| \geq \kappa$ for all $t \in [p]$. For example, in the simple linear logistic regression, we can set $\kappa$ to be proportional to $\min_t\{|w_t|\}$ to satisfy this condition, where $w_1, \ldots, w_p$ are linear coefficients. In addition, this condition actually controls the change rate for the loss function around test samples to be at least $\kappa$, therefore it gives a baseline for measuring the attack power. Once we have the covering number of $\widetilde{\mathcal{F}}_{\mathcal{W}_L}$, it is easy to access an upper bound on $\widehat{\mathfrak{R}}(\widetilde{\mathcal{F}}_{\mathcal{W}_L})$ via Dudley's entropy integral. According to the Ledoux-Talagrand contraction inequality in Theorem 3.2.3, $\widehat{\mathfrak{R}}(g \circ \widetilde{\mathcal{F}}_{\mathcal{W}_L}) \leq \widehat{\mathfrak{R}}(\widetilde{\mathcal{F}}_{\mathcal{W}_L})$ since we assume that the loss function $g(\cdot, y)$ is 1-Lipschitz for a fixed $y$. As a result, we establish the following theorem as the first contribution of our work.

**Theorem 4.2.1.** *Under the same assumptions as in Lemma 4.2.1, the Rademacher complexity of* $g \circ \widetilde{\mathcal{F}}_{\mathcal{W}_L}$ *is upper bounded by*

$$\mathcal{O}\left(\frac{\Delta}{\sqrt{m}}\sqrt{p\sum_{\mathrm{j}=1}^{L} r_\mathrm{j} \ln\left(L\sqrt{m}\left(1+\frac{\Gamma}{B+\sqrt{p}\epsilon}\right)\right)}\right), \tag{4.4}$$

*where* $\Delta = \prod_{\mathrm{j}=1}^{L} c_\mathrm{j}(B+\sqrt{p}\epsilon)$ *and* $\Gamma$ *is given in Lemma 4.2.1.*

*Proof.* By standard Dudley's entropy integral, we have

$$\begin{aligned}
\widehat{\mathfrak{R}}_m(\widetilde{F}_{\mathcal{W}_L}) &\lesssim \inf_{\beta>0}\left\{\beta + \frac{1}{\sqrt{m}}\int_\beta^\alpha \sqrt{\ln N(\widetilde{F}_{\mathcal{W}_L}, \tau, \rho_m)}d\tau\right\} \\
&\leq \inf_{\beta>0}\left\{\beta + \frac{1}{\sqrt{m}}\int_\beta^\alpha \sqrt{(2p+1)\sum_{\mathrm{j}=1}^{L} r_\mathrm{j} \ln\left(\frac{9L(B+\sqrt{p}\epsilon+\Gamma)\prod_{\mathrm{j}=1}^{L} c_\mathrm{j}}{\tau}\right)}d\tau\right\} \\
&\leq \inf_{\beta>0}\left\{\beta + \frac{\alpha}{\sqrt{m}}\sqrt{(2p+1)\sum_{\mathrm{j}=1}^{L} r_\mathrm{j} \ln\left(\frac{9L(B+\sqrt{p}\epsilon+\Gamma)\prod_{\mathrm{j}=1}^{L} c_\mathrm{j}}{\beta}\right)}\right\}
\end{aligned}$$

Here $\alpha = \prod_{\mathrm{j}=1}^{L} c_\mathrm{j}(B+\sqrt{p}\epsilon)$. Take $\beta = \alpha/\sqrt{m}$, we have

$$\widehat{\mathfrak{R}}_m(g \circ \widetilde{F}_{\mathcal{W}_L}) \leq \widehat{\mathfrak{R}}_m(\widetilde{F}_{\mathcal{W}_L}) \leq \mathcal{O}\left(\frac{\prod_{\mathrm{j}=1}^{L} c_\mathrm{j}(B+\sqrt{p}\epsilon)}{\sqrt{m}}\sqrt{p\sum_{\mathrm{j}=1}^{L} r_\mathrm{j} \ln\left(L\sqrt{m}\left(1+\frac{\Gamma}{B+\sqrt{p}\epsilon}\right)\right)}\right).$$

□

**Remark 4.2.1.** *Since logarithm is an order of constant, the upper bound in Theorem 4.2.1 can be simplified as* $\tilde{\mathcal{O}}(\Delta\sqrt{p\sum r_{\mathrm{j}}/m})$. *Hence, it is clear that the effect of adversarial attacks on generalization performance is an additional linear term with* $\epsilon$. *The corresponding linear coefficient includes the width of the neural network, the sum of ranks of weight matrices, and the product of spectral norms of weight matrices, which is exactly the Lipschitz constant of the neural network. Assuming the ranks are all equal to* $r$, $\sum_{\mathrm{j}=1}^{L} r_{\mathrm{j}}$ *turns to be a linear function of the depth, which implies that the depth influences the generalization error of adversarial learning to a certain extent. In the next section, we improve this result and establish a tighter bound, so that the bound achieves depth-free. Note that when* $\epsilon = 0$, *it recovers the case for natural learning. We leave the comparison of risk bounds between natural learning and adversarial learning to Section 4.*

It is worthwhile to compare Theorem 4.2.1 with several existing works. When the DNN $f_{\mathcal{W}_L}$ reduce to a linear classifier, the adversarial example $\tilde{\mathbf{x}}$ produced by FGSM is the exact solution to $\operatorname{argmax}_{\mathbf{x}':\|\mathbf{x}'-\mathbf{x}\|_\infty\leq\epsilon} g(f_{\mathcal{W}_L}(\mathbf{x}'), y)$, which corresponds to the case in [19]. For multi-class linear classification, the bound in [19] relies on the width by an order of $p^2$, while our bound is $\mathcal{O}\Big((B+\sqrt{p}\epsilon)\sqrt{pr_1/m}\Big)$, which is sharper than theirs by a factor of $\sqrt{p}$ if $r_1 = p$. Instead, for binary linear classification, the upper bound in Theorem 4.2.1 reduces to $\mathcal{O}\Big((B+\sqrt{p}\epsilon)\sqrt{p/m}\Big)$. This is comparable to the result in [19] by an additional $\sqrt{p}$-factor. Under similar assumptions, [18] provides an upper bound for the generalization error of DNNs under the FGM, which constrains the attacks within an $\ell_2$-ball. Their result relies on the depth by a factor of $L$ and the width by a factor of $\sqrt{p}$. We focus on the $\ell_\infty$ attacks and thus we believe that these two approaches are not directly comparable. [20] establishes upper bounds on the surrogate tree transform, resting on $\ell_\infty$-operator norms and Frobenius norms of weight matrices. For multi-class neural networks, their bound is polynomially dependent on the depth and the width of the DNN by a factor of $\sqrt{L}$ and $p$, respectively. This adversarial Rademacher complexity upper bound is similar to ours. In the next section, we provide a tighter depth-free upper bound.

### 4.2.2 A Tighter Upper Bound on Rademacher Complexity for Adversarial Learning

As we discussed previously, an $L$-layer neural network $f_{\mathcal{W}_L} : \mathbb{R}^{p_0} \to \mathbb{R}^{p_L} \in \mathcal{F}_{\mathcal{W}_L}$ is a $\prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}}$-Lipschitz continuous function if all activation functions are 1-Lipschitz. It is easy to show that the covering number of DNNs class is independent of the network depth if we regard the whole network as a Lipschitz continuous function. Furthermore, weight matrices tend to be low rank in many empirical results, and drop out can be treated as the low-rank regularization. These take-home points motivate us to decompose $f_{\mathcal{W}_L}$ as a shallow network and a low dimensional Lipschitz continuous function. Relying on this important observation, we establish a tighter upper bound on the Rademacher complexity in adversarial setting, and further limit the effect of adversarial perturbations on the adversarial generalization performance.

Suppose $W_l = U_l \Sigma_l V_l^\top$ for $\forall\ l \in [L]$, where $U_l$ and $V_l$ are column-orthogonal matrices, and $\Sigma_l \in \mathbb{R}^{r_l \times r_l}$ is a diagonal matrix whose entries are non-zero singular values of $W_l$. Then we rewrite $f_{\mathcal{W}_L}(\tilde{\mathbf{x}})$ as

$$f_{\mathcal{W}_L}(\tilde{\mathbf{x}}) = h_{r_l} \circ f_{\mathcal{W}_l}(\tilde{\mathbf{x}}),$$

where

$$f_{\mathcal{W}_l}(\tilde{\mathbf{x}}) = V_l^\top \sigma_{l-1}(\cdots \sigma_1(W_1(\tilde{\mathbf{x}})))$$

is a depth-$l$ neural network and

$$h_{r_l}(\mathbf{z}) = W_L \sigma_{L-1}(W_{L-1} \cdots \sigma_l(U_l \Sigma_l \mathbf{z}))$$

is a Lipschitz continuous function with low dimensional input, mapping from $\mathbb{R}^{r_l}$ to $\mathbb{R}^{p_L}$. The composition implies that $g \circ \widetilde{\mathcal{F}}_{\mathcal{W}_L}$ is a subset of $g \circ \mathcal{H}_{r_l} \circ \widetilde{\mathcal{F}}_{\mathcal{W}_l}$, where

$$\mathcal{H}_{r_l} = \Big\{ \mathbf{z} \mapsto h_{r_l}(\mathbf{z}) \Big| \|\mathbf{z}\| \leq \prod_{\mathrm{j}=1}^{l-1} c_{\mathrm{j}}(B + \sqrt{p}\epsilon),\ \text{Lipschitz constant is } \prod_{\mathrm{j}=l}^{L} c_{\mathrm{j}} \Big\}.$$

According to the properties of Rademacher complexity, we have

$$\widehat{\mathfrak{R}}_m(g \circ \widetilde{\mathcal{F}}_{\mathcal{W}_L}) \leq \widehat{\mathfrak{R}}_m(g \circ \mathcal{H}_{r_l} \circ \widetilde{\mathcal{F}}_{\mathcal{W}_l}).$$

Since this decomposition holds true for any $l \in [L]$, we further obtain the upper bound on $\widehat{\mathfrak{R}}_m(g \circ \widetilde{\mathcal{F}}_{\mathcal{W}_L})$ by choosing the minimum among all $\widehat{\mathfrak{R}}_m(g \circ \mathcal{H}_{r_l} \circ \widetilde{\mathcal{F}}_{\mathcal{W}_l})$ for $l \in [L]$.

To derive an upper bound on $\widehat{\mathfrak{R}}_m(g \circ \mathcal{H}_{r_l} \circ \widetilde{\mathcal{F}}_{\mathcal{W}_l})$, we additionally need the covering number for the class of Lipschitz continuous functions. The following lemma generalized from Theorem 17 of [55] provides the result.

**Lemma 4.2.2.** *Assume that $\widetilde{\mathcal{H}}$ is a class of $M$-Lipschitz continuous functions mapping from $\mathbb{R}^r$ to $\mathbb{R}$. Let $\mathbf{z}_{\mathrm{i}} \in \mathcal{Z} = \{\mathbf{z} \in \mathbb{R}^r : \|\mathbf{z}\| \leq A\}$ for $\mathrm{i} = 1, \ldots, m$. Then, the covering number of $\widetilde{\mathcal{H}}$ with respect to the metric $\rho_\infty$ satisfies*

$$N\left(\widetilde{\mathcal{H}}, \tau, \rho_\infty\right) \leq \left(2\left\lceil \frac{4MA}{\tau} \right\rceil + 1\right)^{\left(\frac{6MA}{\tau}\right)^r}.$$

Notice that Lemma 4.2.1 holds true for DNNs with any number of layers. With the aid of Lemma 4.2.1 and Lemma 4.2.2, next Lemma 4.2.3 provides the Rademacher complexity for $g \circ \mathcal{H}_{r_l} \circ \widetilde{\mathcal{F}}_{\mathcal{W}_l}$ under different assumptions on the rank $r_l$. It is interesting and surprising that different rank constraints lead to different sample complexities for $g \circ \mathcal{H}_{r_l} \circ \widetilde{\mathcal{F}}_{\mathcal{W}_l}$.

**Lemma 4.2.3.** *Under the same assumptions as in Lemma 4.2.1, we have the following three bounds:*

1. *When $r_l = 1$, $\widehat{\mathfrak{R}}_m(g \circ \mathcal{H}_{r_l} \circ \widetilde{\mathcal{F}}_{\mathcal{W}_l})$ satisfies*

$$R_l^1 := \mathcal{O}\left(\frac{\Delta}{\sqrt{m}} \sqrt{p \sum_{\mathrm{j}=1}^{l} r_{\mathrm{j}} \ln\left(l\sqrt{m}\left(1 + \frac{\Gamma}{B + \sqrt{p}\epsilon}\right)\right)}\right), \tag{4.5}$$

*where $\Delta$ and $\Gamma$ are defined in Theorem 4.2.1 and Lemma 4.2.1.*

2. *When* $r_l = 2$, $\widehat{\mathfrak{R}}_m(g \circ \mathcal{H}_{r_l} \circ \widetilde{\mathcal{F}}_{\mathcal{W}_l})$ *satisfies*

$$R_l^2 := \mathcal{O}\left(\frac{\Delta}{\sqrt{m}}\left(\sqrt{(\ln\sqrt{m})^3} + \sqrt{p\sum_{\mathrm{j}=1}^{l} r_{\mathrm{j}} \ln\left(l\sqrt{m}\left(1 + \frac{\Gamma}{B+\sqrt{p}\epsilon}\right)\right)}\right)\right). \tag{4.6}$$

3. *When* $r_l \geq 3$, $\widehat{\mathfrak{R}}_m(g \circ \mathcal{H}_{r_l} \circ \widetilde{\mathcal{F}}_{\mathcal{W}_l})$ *satisfies*

$$R_l^{r_l} := \mathcal{O}\left(\frac{\Delta}{\sqrt[r_l]{m}}\left(\frac{\sqrt{24^{r_l}\ln(\sqrt[r_l]{m})}}{r_l} + \sqrt{p\sum_{\mathrm{j}=1}^{l} r_{\mathrm{j}} \ln\left(l\sqrt[r_l]{m}\left(1 + \frac{\Gamma}{B+\sqrt{p}\epsilon}\right)\right)}\right)\right). \tag{4.7}$$

**Remark 4.2.2.** *When* $r_l \geq 3$, $\widehat{\mathfrak{R}}_m(g \circ \mathcal{H}_{r_l} \circ \widetilde{\mathcal{F}}_{\mathcal{W}_l})$ *exponentially depends on the rank* $r_l$. *Instead, we can choose to use the Rademacher complexity bound of* $g \circ \widetilde{\mathcal{F}}_{\mathcal{W}_L}$ *established in Theorem 4.2.1. Hence, the upper bound for this case should be the minimum of Equation 4.4 and Equation 4.7.*

Consider some special cases to better illustrate results in Lemma 4.2.3. When two weight matrices at layer $l_1$ and $l_2$ with $l_1 < l_2$ have the same rank $r$, Lemma 4.2.3 shows that $R_{l_1}^r < R_{l_2}^r$. Hence, the bound on $\widehat{\mathfrak{R}}_m(g \circ \mathcal{H}_{r_l} \circ \widetilde{\mathcal{F}}_{\mathcal{W}_l})$ can be tighter if the depth $l$ of the shallow network can be as small as possible. On the other hand, if the depth $l$ is fixed, the larger the rank is, the larger the bound on $\widehat{\mathfrak{R}}_m(g \circ \mathcal{H}_{r_l} \circ \widetilde{\mathcal{F}}_{\mathcal{W}_l})$ is. Therefore, the ideal decomposition of a network is to find the smallest layer whose weight matrix produces the smallest rank. To achieve this, we first find the corresponding weight matrix with the smallest depth for all possible ranks ranging from 1 to $p$. Then $\widehat{\mathfrak{R}}_m(g \circ \mathcal{H}_{r_l} \circ \widetilde{\mathcal{F}}_{\mathcal{W}_L})$ is obtained as the minimum among all these initial upper bounds on different ranks. The conclusion is formalized in Theorem 4.2.2.

**Theorem 4.2.2.** *For* $\mathrm{i} = 1, \ldots, p$, *define* $l^{(\mathrm{i})} = \min_{\mathrm{j}\in[L]}\{\mathrm{j} : \mathrm{rank}(W_{\mathrm{j}}) = \mathrm{i}\}$. *Then,* $\widehat{\mathfrak{R}}_m(g \circ \widetilde{\mathcal{F}}_{\mathcal{W}_L}) \leq \min_{\mathrm{i}\in[p]} R_{l^{(\mathrm{i})}}^{\mathrm{i}}$, *where the* $R_{l^{(\mathrm{i})}}^{\mathrm{i}}$ *is defined in Lemma 4.2.3.*

*Proof.* The conclusion of this theorem is a natural consequence of Lemma 4.2.3. □

Theorem 4.2.2 considers all possible decomposition of $g \circ \widetilde{\mathcal{F}}_{\mathcal{W}_L}$ and chooses the optimal one. It provides a new and tighter bound on the Rademacher complexity for adversarial learning assuming the spectral norm and the rank constraints on the weight matrix at each layer. It is interesting that this upper bound is the minimum of $p$ bounds, which may suggest that DNNs behave like an ensemble.

**Remark 4.2.3.** *We ignore logarithmic factors for simplicity and consider two different scenarios. The first case is that there are low rank matrices with rank 1 or 2 at layer $l$. Theorem 4.2.2 shows that the upper bound on the Rademacher complexity of $g \circ \widetilde{\mathcal{F}}_{\mathcal{W}_L}$ is at most of order $\Delta\sqrt{p\sum_{\mathrm{j}=1}^{l} r_{\mathrm{j}}/m}$. The second case is that the ranks of weight matrices are all greater than or equal to 3. Theorem 4.2.2 shows that the upper bound is at most of order*

$$\Delta \cdot \min\left(\sqrt{\frac{p\sum_{\mathrm{j}=1}^{L} r_{\mathrm{j}}}{m}}, \frac{\sqrt{p\sum_{\mathrm{j}=1}^{l} r_{\mathrm{j}}} + \sqrt{24^{r_l}}/r_l}{\sqrt[r_l]{m}}\right).$$

*These bounds are depth free, depend on the Lipschitz constant $\prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}}$, and has a linear relationship with $\epsilon$ whose coefficient is linear in the width $p$.*

Comparing these new bounds with Equation 4.4, Theorem 4.2.2 establishes a tighter bound. To be more precise, the Rademacher complexity of DNNs under adversarial setting only relies on the shallow part if a low rank weight matrix exists. Correspondingly, the linear coefficient of adversarial perturbations reduces to $\mathcal{O}(\sqrt{\sum_{\mathrm{j}=1}^{l} r_{\mathrm{j}}})$, which indicates that adversarial attacks have a smaller influence on the Rademacher complexity. As we discussed in the previous section, current existing works analyze the adversarial generalization bound for linear classifier and surrogate of DNNs. Their bounds polynomially depends on the depth and the width. However, we work on the neural network space directly and the bound is depth free and linear in $\epsilon$.

## 4.3 Natural Learning vs. Adversarial Learning

To compare the generalization behaviors between natural learning and adversarial learning, it would be more convenient if the Rademacher complexity for natural learning can

be computed explicitly. Corollary 4.3.1 and Corollary 4.3.2 establish upper bounds on the Rademacher complexity of $g \circ \mathcal{F}_{\mathcal{W}_L}$ by setting $\epsilon = 0$ in Theorem 4.2.1 and Lemma 4.2.3.

**Corollary 4.3.1.** *Assume the activation function $\sigma(\cdot)$ is 1-Lipschitz, satisfying $\sigma(\mathbf{0}) = \mathbf{0}$, and the loss function $g(\cdot, y)$ is 1-Lipschitz for any fixed label $y$. Let $\mathbf{x}_{\mathrm{i}} \in \mathcal{X} = \{\mathbf{x} \in \mathbb{R}^p : \left\|\mathbf{x}\right\| \leq B\}$ for $\mathrm{i} = 1, \ldots, m$. Then the Rademacher complexity $\widehat{\mathfrak{R}}_m(g \circ \mathcal{F}_{\mathcal{W}_L})$ for natural learning is upper bounded by*

$$\mathcal{O}\left(\frac{\prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}} B}{\sqrt{m}} \sqrt{p \sum_{\mathrm{j}=1}^{L} r_{\mathrm{j}} \ln\left(L\sqrt{m}\right)}\right).$$

The bound obtained by Corollary 4.3.1 recovers preexisting natural learning risk bounds [49]. Similar to Lemma 4.2.3, we provide a tighter upper bound for $\widehat{\mathfrak{R}}(g \circ \mathcal{F}_{\mathcal{W}_L})$ by realizing a DNN as the decomposition of a low-dimensional Lipschitz continuous function and a shallow network.

**Corollary 4.3.2.** *Under the same assumptions as in Corollary 4.3.1, define $\Delta = \prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}} B$ and $\Gamma = 0$. The bounds of $\widehat{\mathfrak{R}}_m(g \circ \mathcal{H}_{r_l} \circ \mathcal{F}_{\mathcal{W}_l})$ are obtained by plugging $\Delta$ and $\Gamma$ in Equation 4.5, Equation 4.6, and Equation 4.7 respectively.*

Again, we take the bound for $\widehat{\mathfrak{R}}_m(g \circ \mathcal{F}_{\mathcal{W}_L})$ as the minimum among all $\widehat{\mathfrak{R}}_m(g \circ \mathcal{H}_{r_l} \circ \mathcal{F}_{\mathcal{W}_l})$, so Theorem 4.2.2 can be applied here directly.

This new Rademacher complexity bound for natural learning has its own independent interests, which is the tightest upper bound in the literature [44]–[50]. For example, [44] provides a generalization error exponentially depending on depth $L$. [47] shows a bound based on spectral norm and $L_{2,1}$ norm, but this bound is still polynomial dependent on depth $L$. [45] also establishes a bound polynomially depending on depth $L$ and width $p$. [49] achieves a bound scaling as $\mathcal{O}\left(B \prod_{\mathrm{j}=1}^{L} \left\|W_{\mathrm{j}}\right\|_2 \sqrt{Lpr/m}\right)$ by assuming all ranks are the same. Our bound reach an order of $\mathcal{O}\left(B \prod_{\mathrm{j}=1}^{L} \left\|W_{\mathrm{j}}\right\|_2 \sqrt{p \sum_{\mathrm{j}=1}^{l} r_{\mathrm{j}}/m}\right)$, which is apparently tighter than current existing bounds.

**Remark 4.3.1.** *For ease of illustration, we ignore the logarithm term in the following. Assume there are weight matrices with rank at most 2 at layer $l$, we have*

$$\widehat{\mathfrak{R}}_m^{(U)}(g \circ \widetilde{\mathcal{F}}_{\mathcal{W}_L}) - \widehat{\mathfrak{R}}_m^{(U)}(g \circ \mathcal{F}_{\mathcal{W}_L}) = \mathcal{O}\left(\frac{\prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}} \epsilon p}{\sqrt{m}} \sqrt{\sum_{\mathrm{j}=1}^{l} r_{\mathrm{j}}}\right),$$

*where $\widehat{\mathfrak{R}}_m^{(U)}(g \circ \widetilde{\mathcal{F}}_{\mathcal{W}_L})$ and $\widehat{\mathfrak{R}}_m^{(U)}(g \circ \mathcal{F}_{\mathcal{W}_L})$ denote the upper bound of $\widehat{\mathfrak{R}}_m(g \circ \widetilde{\mathcal{F}}_{\mathcal{W}_L})$ and $\widehat{\mathfrak{R}}_m(g \circ \mathcal{F}_{\mathcal{W}_L})$ respectively. If there are no low-rank matrices, we have*

$$\widehat{\mathfrak{R}}_m^{(U)}(g \circ \widetilde{\mathcal{F}}_{\mathcal{W}_L}) - \widehat{\mathfrak{R}}_m^{(U)}(g \circ \mathcal{F}_{\mathcal{W}_L}) = \mathcal{O}\left(\frac{\prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}} \epsilon p}{\sqrt{m}} \sqrt{\sum_{\mathrm{j}=1}^{L} r_{\mathrm{j}}}\right).$$

*This confirms that the adversarial Rademacher complexity is always as large as its natural counterpart with an additional term $\mathcal{O}(\epsilon/\sqrt{m})$, which infers that it could be larger than the natural Rademacher complexity. The gap between these two Rademacher complexities can be further limited if there exists a low-rank weight matrix in the DNN. This is consistent with the previous conclusion that the low-rank structure is able to reduce the effect of adversarial attacks on generalization error.*

The trade-off between robustness and natural accuracy has been consistently reported in the literature [4], [15]. Training models to be robust may lead to a reduction of standard natural accuracy. The general study on this topic is beyond the scope of this paper. To achieve both robustness and high natural accuracy, [56] proposed to learn the robust model by using both natural examples and adversarial examples during the training process. For samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ from $\mathcal{X}$, we choose a subset of the training examples to create adversarial examples, denoted as $\mathcal{S}_a$, while the remaining subset is natural examples, denoted as $\mathcal{S}_n$. A new loss function that can independently control the size of adversarial examples is defined as

$$\frac{1}{m}\left(\sum_{\mathbf{x}_{\mathrm{i}} \in \mathcal{S}_n} g(f_{\mathcal{W}_L}(\mathbf{x}_{\mathrm{i}}), y_{\mathrm{i}}) + \sum_{\tilde{\mathbf{x}}_{\mathrm{i}} \in \mathcal{S}_a} g(f_{\mathcal{W}_L}(\tilde{\mathbf{x}}_{\mathrm{i}}), y_{\mathrm{i}})\right). \tag{4.8}$$

We then define the function class $G \circ \mathcal{F}_{\mathcal{W}_L} = \{(\mathbf{x}, y) \mapsto G(f_{\mathcal{W}_L}(\mathbf{x}), y)\}$ for this new loss function, where

$$\begin{aligned} & G(f_{\mathcal{W}_L}(\mathbf{x}), y) \\ =& g(f_{W_L}(\mathbf{x}), y)\mathbb{I}(\mathbf{x} \in \mathcal{S}_n) + g(f_{W_L}(\tilde{\mathbf{x}}), y)\mathbb{I}(\tilde{\mathbf{x}} \in \mathcal{S}_a) \end{aligned}$$

An upper bound on the Rademacher complexity for $G \circ \mathcal{F}_{\mathcal{W}_L}$ is established in Theorem 4.3.1.

**Theorem 4.3.1.** *Given $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ from $\mathcal{X}$, let $\mathcal{S}_a$ be the adversarial set and $\mathcal{S}_n$ be the natural set. The Rademacher complexity of $G \circ \mathcal{F}_{\mathcal{W}_L}$ satisfies*

$$\widehat{\mathfrak{R}}_m(G \circ \mathcal{F}_{\mathcal{W}_L}) \leq \frac{|\mathcal{S}_n|}{m}\widehat{\mathfrak{R}}_{|\mathcal{S}_n|}(g \circ \mathcal{F}_{\mathcal{W}_L}) + \frac{|\mathcal{S}_a|}{m}\widehat{\mathfrak{R}}_{|\mathcal{S}_a|}(g \circ \widetilde{\mathcal{F}}_{\mathcal{W}_L}),$$

*where $\widehat{\mathfrak{R}}_{|\mathcal{S}_n|}(g \circ \mathcal{F}_{\mathcal{W}_L})$ and $\widehat{\mathfrak{R}}_{|\mathcal{S}_a|}(g \circ \widetilde{\mathcal{F}}_{\mathcal{W}_L})$ are the Rademacher complexities computed on data samples in $\mathcal{S}_n$ and $\mathcal{S}_a$ respectively.*

*Proof.* Following the definition, we have

$$\begin{aligned} \widehat{\mathfrak{R}}_m(G \circ \mathcal{F}_{\mathcal{W}_L}) &= \mathbb{E}_\delta\left[\sup_{G \circ f_{\mathcal{W}_L} \in G \circ \mathcal{F}_{\mathcal{W}_L}} \frac{1}{m}\sum_{\mathrm{i}=1}^{m} \delta_{\mathrm{i}}\Big(g(f_{W_L}(\mathbf{x}_{\mathrm{i}}), y_{\mathrm{i}})\mathbb{I}(\mathbf{x}_{\mathrm{i}} \in \mathcal{S}_n) + g(f_{W_L}(\tilde{\mathbf{x}}_{\mathrm{i}}), y_{\mathrm{i}})\mathbb{I}(\tilde{\mathbf{x}}_{\mathrm{i}} \in \mathcal{S}_a)\Big)\right] \\ &\leq \mathbb{E}_\delta\left[\sup_{g \circ f_{\mathcal{W}_L} \in g \circ \mathcal{F}_{\mathcal{W}_L}} \frac{1}{m}\sum_{\mathrm{i}=1}^{m} \delta_{\mathrm{i}}\Big(g(f_{W_L}(\mathbf{x}_{\mathrm{i}}), y_{\mathrm{i}})\mathbb{I}(\mathbf{x}_{\mathrm{i}} \in \mathcal{S}_n)\Big)\right] \\ &+ \mathbb{E}_\delta\left[\sup_{g \circ f_{\mathcal{W}_L} \in g \circ \widetilde{F}_{\mathcal{W}_L}} \frac{1}{m}\sum_{\mathrm{i}=1}^{m} \delta_{\mathrm{i}}\Big(g(f_{W_L}(\tilde{\mathbf{x}}_{\mathrm{i}}), y_{\mathrm{i}})\mathbb{I}(\tilde{\mathbf{x}}_{\mathrm{i}} \in \mathcal{S}_a)\Big)\right] \\ &= \frac{|\mathcal{S}_n|}{m}\mathbb{E}_\delta\left[\sup_{g \circ f_{\mathcal{W}_L} \in g \circ \mathcal{F}_{\mathcal{W}_L}} \frac{1}{|\mathcal{S}_n|}\sum_{\mathbf{x}_{\mathrm{i}} \in \mathcal{S}_n} \delta_{\mathrm{i}} g(f_{W_L}(\mathbf{x}_{\mathrm{i}}), y_{\mathrm{i}})\right] \\ &+ \frac{|\mathcal{S}_a|}{m}\mathbb{E}_\delta\left[\sup_{g \circ f_{\mathcal{W}_L} \in g \circ \widetilde{\mathcal{F}}_{\mathcal{W}_L}} \frac{1}{|\mathcal{S}_a|}\sum_{\tilde{x}_{\mathrm{i}} \in \mathcal{S}_a} \delta_{\mathrm{i}} g(f_{W_L}(\tilde{\mathbf{x}}_{\mathrm{i}}), y_{\mathrm{i}})\right] \\ &= \frac{|\mathcal{S}_n|}{m}\widehat{\mathfrak{R}}_{|\mathcal{S}_n|}(g \circ \mathcal{F}_{\mathcal{W}_L}) + \frac{|\mathcal{S}_a|}{m}\widehat{\mathfrak{R}}_{|\mathcal{S}_a|}(g \circ \widetilde{\mathcal{F}}_{\mathcal{W}_L}). \end{aligned}$$

This completes the proof. □

Theorem 4.3.1 shows that the Rademacher complexity bound for $G \circ \mathcal{F}_{\mathcal{W}_L}$ is a linear combination between the Rademacher complexities computed on natural examples and adversarial examples. The coefficients are related to the sample sizes $|\mathcal{S}_n|$ and $|\mathcal{S}_a|$. If the

training is based on the loss function (Equation 4.8), Theorem 4.3.1 confirmed that controlling the sample sizes $|\mathcal{S}_n|$ and $|\mathcal{S}_a|$ can contribute to the trade-off between robustness and natural accuracy.

## 4.4 Numerical Results

In this section, we provide a set of experiments to validate our theoretical findings. In particular, we show 1) the spectral normalization reduces the gap between the population adversarial risk and empirical adversarial risk; 2) the adversarial generalization bound is depth-free if there exists a low-rank weight matrix; 3) the generalization error is proportional to the attack size.

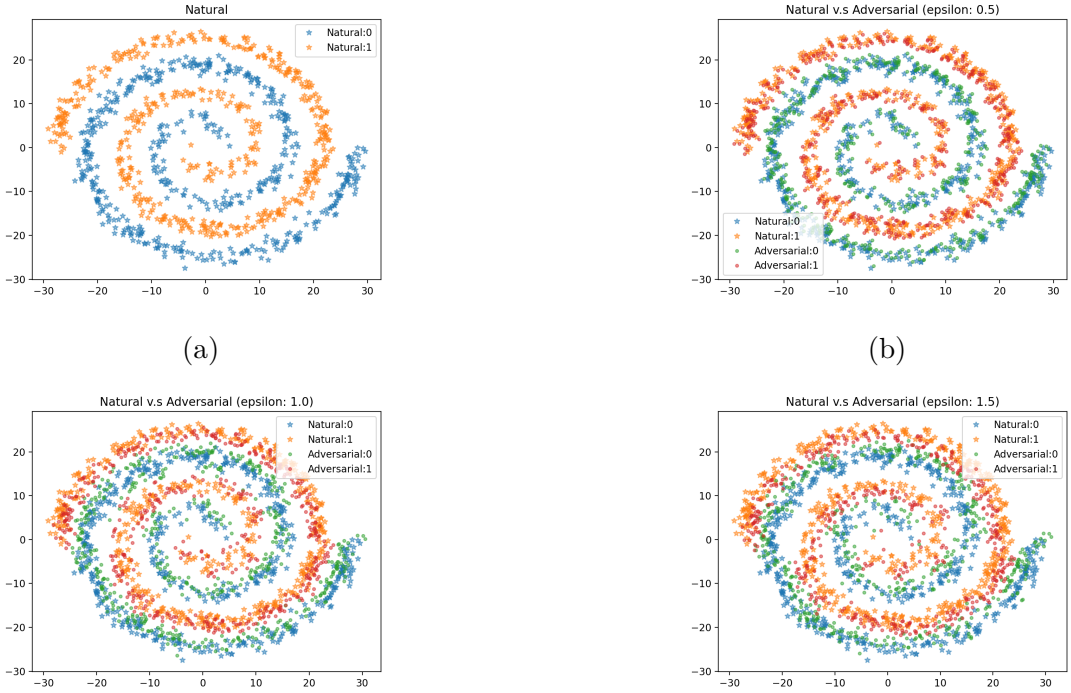



**Figure 4.1.** Original testing data and adversarial testing data: (a) Original data with label 0 and label 1; (b), (c), (d) Original data with label 0 and label 1 are perturbed by the FGSM scheme that uses the model NN-5-10-2 with spectral normalization, and the attack size is 0.5, 1.0 and 1.5 respectively. It becomes harder to separate the data as the attack size increases.

**Table 4.1.** Adversarial generalization errors under FGSM attacks for various model structures with different constraints on weight matrices.

| Model | | $\epsilon = 0.0$ | $\epsilon = 0.5$ | $\epsilon = 1.0$ | $\epsilon = 1.5$ |
|---|---|---|---|---|---|
| With SN | NN-5-10-2 | 0.0012 | 0.0128 | 0.0234 | 0.0332 |
| | NN-5-10-2-5 | 0.0042 | 0.0096 | 0.0204 | 0.0364 |
| | NN-5-10-2-5-8 | 0.0028 | 0.0090 | 0.0222 | 0.0320 |
| Without SN | NN-5-10-2 | 0.0062 | 0.0156 | 0.0308 | 0.0404 |
| | NN-5-10-2-5 | 0.0068 | 0.0158 | 0.0230 | 0.0302 |
| | NN-5-10-2-5-8 | 0.0056 | 0.0114 | 0.0340 | 0.0386 |

We consider the simple two-spirals dataset demonstrated in Figure 4.1a, where the training set includes 5000 samples and testing set has 1000 samples. Several network structures are constructed to verify our depth-free conclusion. For convenience, we denote a neural network with $L-1$ hidden layers as NN-$p_1$-$\cdots$-$p_{L-1}$. Because this is a binary classification problem and the dimension of the input is 2, both the input layer and output layer have 2 units and we omit it in the notation. The FGSM attack adversarial training is applied to minimize the empirical adversarial risk in Equation 1.1. In each iteration, we first take the gradient of current loss on clean data to generate adversarial data, then update the model with the adversarial data. Once the model is obtained, we use FGSM attack to check the adversarial training and testing error. The attack sizes are set as 0.5, 1.0 and 1.5 respectively. We also constrain the spectral norm of every weight matrix to be 1. Under each setting, we train the model for 150 epochs using Adam with a batch size of 50, and the learning rate is 0.001. Figure 4.1b, Figure 4.1c and Figure 4.1d plot adversarial examples generated by model NN-5-10-2 under the FGSM attack. It is obvious that the larger the attack size is, the more difficult it is to separate these two spirals.

We present the adversarial generalization errors under the FGSM attack in Table 4.1. As we can see, the generalization error for the model with spectral normalization is generally smaller than that for the model without spectral normalization under the same attack power. Thus constraining spectral norms of weight matrices indeed reduce the adversarial generalization error under the FGSM attack.

To check whether the adversarial generalization error for FGSM is depth-free under constraints of spectral normalization and low-rank weight matrices, we construct three networks

NN-5-10-2, NN-5-10-2-5, and NN5-10-2-5-8. They have different numbers of hidden layers, but all models have one two-units layer. For each attack size, it is clear that generalization errors of these three models are similar, which indicates that the generalization error only relies on the small shallow network and is independent on the depth of the whole network.

In terms of the relationship between generalization error and attack size, we can tell that the adversarial generalization errors increase as the attack size increases regardless of whether the spectral normalization is adapted. However, the linearity between these two terms are more significant for models with spectral normalization. Overall, our experiments confirms that applying spectral norm and low rank regularization can improve the generalization behavior of adversarial learning.

## 4.5 Related Proofs

### 4.5.1 Proof of Lemma 4.2.1

**Lemma 4.5.1.** *Let* $S_{c,r} = \{W : W \in \mathbb{R}^{p_2 \times p_1}, \left\|W\right\|_2 \leq c, \mathrm{rank}(W) \leq r\}$. *Then there exists an* $\tau$*-covering of* $S_{c,r}$ *with respect to the spectral norm obeying*

$$N(S_{c,r}, \tau, \left\| \cdot \right\|_2) \leq \left(\frac{9c}{\tau}\right)^{r(p_2+p_1+1)}.$$

*Proof.* We prove the lemma by extending the arguments from [57]. We do SVD of $W$ in $S_{c,r}$,

$$W = U\Sigma V^\top = cU\frac{\Sigma}{c}V^\top := cU\tilde{\Sigma}V^\top, \tag{4.9}$$

where $\Sigma \in \mathbb{R}^{r\times r}$ is the diagonal matrix with singular values, $U \in \mathbb{R}^{p_2\times r}$ and $V \in \mathbb{R}^{p_1\times r}$ are column orthogonal matrices. Thus, $\|U\|_2 = \|V\|_2 = 1$, and $\|\tilde{\Sigma}\|_2 \leq 1$. We will construct an $\tau$-covering for $S_{c,r}$ by covering the set of $U$, $\tilde{\Sigma}$ and $V$. we assume $p_1 = p_2 = p$ for simplicity.

Let $\Lambda$ be the set of diagonal matrices with non-negative entries and spectral norm less than 1. We take $\Lambda'$ to be an $\tau/(3c)$-net for $\Lambda$ with

$$|\Lambda'| \leq \left(\frac{9c}{\tau}\right)^r.$$

Let $O_{p,r} = \{U \in \mathbb{R}^{p\times r} : \left\|U\right\|_2 = 1\}$. There also exists an $\tau/(3c)$-net $O'_{p,r}$ for $O_{p,r}$ obeying

$$|O'_{p,r}| \leq \left(\frac{9c}{\tau}\right)^{pr}.$$

We now let $S'_{c,r} = \{cU'\Sigma'V'^{\top} : U', V' \in O'_{p,r}, \Sigma' \in \Lambda'\}$.Thus,

$$|S'_r| \leq |O'_{p,r}|^2|\Lambda'| \leq \left(\frac{9c}{\tau}\right)^{r(2p+1)}.$$

It remains to show that there exists $S'_{c,r}$ for $S_{c,r}$, such that $\left\|W - W'\right\|_2 \leq \tau$.

$$\begin{aligned}
&\left\|W - W'\right\|_2 \\
=&c\left\|U\tilde{\Sigma}V^{\top} - U'\Sigma'V'^{\top}\right\|_2 \\
=&c\left\|U\tilde{\Sigma}V^{\top} - U'\tilde{\Sigma}V^{\top} + U'\tilde{\Sigma}V^{\top} - U'\Sigma'V^{\top} + U'\Sigma'V^{\top} - U'\Sigma'V'^{\top}\right\|_2 \\
\leq&c\Big(\left\|(U-U')\tilde{\Sigma}V^{\top}\right\|_2 + \left\|U'(\tilde{\Sigma}-\Sigma')V^{\top}\right\|_2 + \left\|U'\Sigma'(V-V')^{\top}\right\|_2\Big)
\end{aligned}$$

For the first term,

$$\left\|(U-U')\tilde{\Sigma}V^{\top}\right\|_2 \leq \left\|U-U'\right\|_2\left\|\tilde{\Sigma}\right\|_2\left\|V\right\|_2 \leq \frac{\tau}{3c}.$$

The same argument gives $\left\|U'\Sigma'(V-V')^{\top}\right\|_2 \leq \tau/(3c)$. For the second term,

$$\left\|U'(\tilde{\Sigma}-\Sigma')V^{\top}\right\|_2 \leq \left\|\tilde{\Sigma}-\Sigma'\right\|_2 \leq \frac{\tau}{3c}.$$

Therefore, $\left\|W - W'\right\|_2 \leq \tau$. This completes the proof. □

**Lemma 4.5.2.** *Given $(\mathbf{x}_1, \ldots, \mathbf{x}_m)$ from $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^p : \left\|\mathbf{x}\right\| \leq B\}$, each $\mathbf{x}_{\mathrm{i}}$ is perturbed by $\epsilon \cdot sign(\nabla_{\mathbf{x}_{\mathrm{i}}} g(f_{\mathcal{W}_L}(\mathbf{x}_{\mathrm{i}}), y_{\mathrm{i}}))$. Assume the activation function $\sigma(\cdot)$ is 1-Lipschitz and 1-smooth. Assume the loss function $g(\cdot, y)$ is 1-Lipschitz and 1-smooth for any fixed label $y$, and $\min_{t\in[p]} |\nabla^{(t)}_{\mathbf{x}_{\mathrm{i}}} g(f_{\mathcal{W}_L}(\mathbf{x}_{\mathrm{i}}), y_{\mathrm{i}})| \geq \kappa$ holds for a constant $\kappa > 0$, where $\nabla^{(t)}_{\mathbf{x}_{\mathrm{i}}} g(f_{\mathcal{W}_L}(\mathbf{x}_{\mathrm{i}}), y_{\mathrm{i}})$ is the $t$-th*

*element of* $\nabla_{\mathbf{x}_\mathrm{i}} g(f_{\mathcal{W}_L}(\mathbf{x}_\mathrm{i}), y_\mathrm{i})$. *Given* $\mathcal{W}_L = (W_1, \ldots, W_L)$, *there exists* $\mathcal{W}'_L = (W'_1, \ldots, W'_L)$, *where* $W'_\mathrm{j}$ *is from the* $\tau_\mathrm{j}$*-covering of* $S_{c_\mathrm{j}, r_\mathrm{j}}$, *for* $\mathrm{j} = 1, \ldots, L$. *Then*

$$\left\|\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}), y) - \nabla_{\mathbf{x}} g(f_{\mathcal{W}'_L}(\mathbf{x}), y)\right\|_2 \leq \prod_{\mathrm{j}=1}^{L} c_\mathrm{j} \sum_{\mathrm{j}=1}^{L} \left(\frac{\tau_\mathrm{j}}{c_\mathrm{j}} + B \prod_{k=1}^{\mathrm{j}} c_k \sum_{k=1}^{\mathrm{j}} \frac{\tau_k}{c_k}\right).$$

*Proof.* According to the Lipschitz and smooth assumptions of the activation function and loss function, we have

$$\begin{aligned}
&\left\|\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}), y) - \nabla_{\mathbf{x}} g(f_{\mathcal{W}'_L}(\mathbf{x}), y)\right\|_2 \\
\leq&\left\|\nabla_{\mathbf{x}} f_{\mathcal{W}_L}(\mathbf{x})(\nabla g)(f_{\mathcal{W}_L}(\mathbf{x}), y)) - \nabla_{\mathbf{x}} f_{\mathcal{W}'_L}(\mathbf{x})(\nabla g)(f_{\mathcal{W}_L}(\mathbf{x}), y))\right\|_2 \\
+&\left\|\nabla_{\mathbf{x}} f_{\mathcal{W}'_L}(\mathbf{x})(\nabla g)(f_{\mathcal{W}_L}(\mathbf{x}), y)) - \nabla_{\mathbf{x}} f_{\mathcal{W}'_L}(\mathbf{x})(\nabla g)(f_{\mathcal{W}'_L}(\mathbf{x}), y))\right\|_2 \\
\leq&\left\|\nabla_{\mathbf{x}} f_{\mathcal{W}_L}(\mathbf{x}) - \nabla_{\mathbf{x}} f_{\mathcal{W}'_L}(\mathbf{x})\right\|_2 + \left\|\nabla_{\mathbf{x}} f_{\mathcal{W}'_L}(\mathbf{x})\right\|_2 \left\|(\nabla g)(f_{\mathcal{W}_L}(\mathbf{x}), y)) - (\nabla g)(f_{\mathcal{W}'_L}(\mathbf{x}), y))\right\|_2 \\
\leq&\left\|\nabla_{\mathbf{x}} f_{\mathcal{W}_L}(\mathbf{x}) - \nabla_{\mathbf{x}} f_{\mathcal{W}'_L}(\mathbf{x})\right\|_2 + \left\|\nabla_{\mathbf{x}} f_{\mathcal{W}'_L}(\mathbf{x})\right\|_2 \left\|f_{\mathcal{W}_L}(\mathbf{x}) - f_{\mathcal{W}'_L}(\mathbf{x})\right\|_2 \qquad (4.10) \\
\leq&\left\|\nabla_{\mathbf{x}} f_{\mathcal{W}_L}(\mathbf{x}) - \nabla_{\mathbf{x}} f_{\mathcal{W}'_L}(\mathbf{x})\right\|_2 + \prod_{\mathrm{j}=1}^{L} c_\mathrm{j}^2 B \sum_{\mathrm{j}=1}^{L} \frac{\tau_\mathrm{j}}{c_\mathrm{j}}. \qquad (4.11)
\end{aligned}$$

Note that Equation 4.10 holds because $g(\cdot)$ is 1-smooth, and

$$\left\|f_{\mathcal{W}_L}(\mathbf{x}) - f_{\mathcal{W}'_L}(\mathbf{x})\right\|_2 \leq \prod_{\mathrm{j}=1}^{L} c_\mathrm{j} B \sum_{\mathrm{j}=1}^{L} \tau_\mathrm{j}/c_\mathrm{j}$$

in Equation 4.11 can be obtained from [47]. Next, we prove the following inequality by induction.

$$\left\|\nabla_{\mathbf{x}} f_{\mathcal{W}_L}(\mathbf{x}) - \nabla_{\mathbf{x}} f_{\mathcal{W}'_L}(\mathbf{x})\right\|_2 \leq \prod_{\mathrm{j}=1}^{L} c_\mathrm{j} \sum_{\mathrm{j}=1}^{L} \left(\frac{\tau_\mathrm{j}}{c_\mathrm{j}} + B \prod_{k=1}^{\mathrm{j}-1} c_k \sum_{k=1}^{\mathrm{j}-1} \frac{\tau_k}{c_k}\right). \qquad (4.12)$$

When $L = 0$, $f_{\mathcal{W}_0}(\mathbf{x}) = \mathbf{x}$, $\left\|\nabla_{\mathbf{x}} f_{\mathcal{W}_0}(\mathbf{x}) - \nabla_{\mathbf{x}} f_{\mathcal{W}'_0}(\mathbf{x})\right\|_2 = 0$. Assume Equation 4.12 holds when there are $L - 1$ layers for DNN. Then, we have

$$
\begin{aligned}
&\left\|\nabla_{\mathbf{x}} f_{\mathcal{W}_L}(\mathbf{x}) - \nabla_{\mathbf{x}} f_{\mathcal{W}'_L}(\mathbf{x})\right\|_2 \\
=&\left\|\nabla_{\mathbf{x}} f_{\mathcal{W}_{L-1}}(\mathbf{x})(\nabla\sigma_{L-1})(f_{\mathcal{W}_{L-1}}(\mathbf{x}))W_L^\top - \nabla_{\mathbf{x}} f_{\mathcal{W}'_{L-1}}(\mathbf{x})(\nabla\sigma_{L-1})(f_{\mathcal{W}'_{L-1}}(\mathbf{x}))W_L'^\top\right\|_2 \\
\leq&\left\|\nabla_{\mathbf{x}} f_{\mathcal{W}_{L-1}}(\mathbf{x})(\nabla\sigma_{L-1})(f_{\mathcal{W}_{L-1}}(\mathbf{x}))W_L^\top - \nabla_{\mathbf{x}} f_{\mathcal{W}_{L-1}}(\mathbf{x})(\nabla\sigma_{L-1})(f_{\mathcal{W}'_{L-1}}(\mathbf{x}))W_L^\top\right\|_2 \\
+&\left\|\nabla_{\mathbf{x}} f_{\mathcal{W}_{L-1}}(\mathbf{x})(\nabla\sigma_{L-1})(f_{\mathcal{W}'_{L-1}}(\mathbf{x}))W_L^\top - \nabla_{\mathbf{x}} f_{\mathcal{W}_{L-1}}(\mathbf{x})(\nabla\sigma_{L-1})(f_{\mathcal{W}'_{L-1}}(\mathbf{x}))W_L'^\top\right\|_2 \\
+&\left\|\nabla_{\mathbf{x}} f_{\mathcal{W}_{L-1}}(\mathbf{x})(\nabla\sigma_{L-1})(f_{\mathcal{W}'_{L-1}}(\mathbf{x}))W_L'^\top - \nabla_{\mathbf{x}} f_{\mathcal{W}'_{L-1}}(\mathbf{x})(\nabla\sigma_{L-1})(f_{\mathcal{W}'_{L-1}}(\mathbf{x}))W_L'^\top\right\|_2 \\
\leq&\prod_{\mathrm{j}=1}^{L-1} c_{\mathrm{j}}\left\|f_{\mathcal{W}_{L-1}}(\mathbf{x}) - f_{\mathcal{W}'_{L-1}}(\mathbf{x})\right\|_2 c_L + \prod_{\mathrm{j}=1}^{L-1} c_{\mathrm{j}}\tau_{\mathrm{j}} + \left\|\nabla_{\mathbf{x}} f_{\mathcal{W}_{L-1}}(\mathbf{x}) - \nabla_{\mathbf{x}} f_{\mathcal{W}'_{L-1}}(\mathbf{x})\right\|_2 c_L \\
\leq&\prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}} \prod_{\mathrm{j}=1}^{L-1} c_{\mathrm{j}} B \sum_{\mathrm{j}=1}^{L-1} \frac{\tau_{\mathrm{j}}}{c_{\mathrm{j}}} + \prod_{\mathrm{j}=1}^{L-1} c_{\mathrm{j}}\tau_{\mathrm{j}} + \prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}} \sum_{\mathrm{j}=1}^{L-1}\left(\frac{\tau_{\mathrm{j}}}{c_{\mathrm{j}}} + B\prod_{k=1}^{\mathrm{j}-1} c_k \sum_{k=1}^{\mathrm{j}-1}\frac{\tau_k}{c_k}\right) \\
\leq&\prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}} \sum_{\mathrm{j}=1}^{L}\left(\frac{\tau_{\mathrm{j}}}{c_{\mathrm{j}}} + B\prod_{k=1}^{\mathrm{j}-1} c_k \sum_{k=1}^{\mathrm{j}-1}\frac{\tau_k}{c_k}\right)
\end{aligned}
$$

Combining Equation 4.11 and Equation 4.12, we can get the conclusion. □

**Lemma 4.5.3.** *Under the same assumptions of Lemma 4.5.2, we have*

$$
\left\|sign(\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}), y)) - sign(\nabla_{\mathbf{x}} g(f_{\mathcal{W}'_L}(\mathbf{x}), y))\right\|_2 \leq \frac{1}{\kappa}(1+\frac{1}{\kappa}\prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}})\prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}} \sum_{\mathrm{j}=1}^{L}\left(\frac{\tau_{\mathrm{j}}}{c_{\mathrm{j}}} + B\prod_{k=1}^{\mathrm{j}} c_k \sum_{k=1}^{\mathrm{j}}\frac{\tau_k}{c_k}\right)
$$

*Proof.* According to the definition,

$$
\mathrm{sign}(\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}), y)) = \Phi^{-1}\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}), y),
$$

where

$$
\Phi = \begin{pmatrix}
|\nabla_{\mathbf{x}}^{(1)} g(f_{\mathcal{W}_L}(\mathbf{x}), y)| & & & \\
& |\nabla_{\mathbf{x}}^{(2)} g(f_{\mathcal{W}_L}(\mathbf{x}), y)| & & \\
& & \ddots & \\
& & & |\nabla_{\mathbf{x}}^{(p)} g(f_{\mathcal{W}_L}(\mathbf{x}), y)|
\end{pmatrix},
$$

in which $\nabla_{\mathbf{x}}^{(t)} g(f_{\mathcal{W}_L}(\mathbf{x}), y)$ is the $t$-th element of $\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}), y)$. Similarly, define $\Phi' = diag(|\nabla_{\mathbf{x}} g(f_{\mathcal{W}'_L}(\mathbf{x}), y)|)$. Then, for fixed $y$,

$$
\begin{aligned}
&\left\|\text{sign}(\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}), y)) - \text{sign}(\nabla_{\mathbf{x}} g(f_{\mathcal{W}'_L}(\mathbf{x}), y))\right\|_2 \\
=&\left\|\Phi^{-1}\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}), y) - (\Phi')^{-1}\nabla_{\mathbf{x}} g(f_{\mathcal{W}'_L}(\mathbf{x}), y)\right\|_2 \\
\leq&\left\|\Phi^{-1}\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}), y) - \Phi^{-1}\nabla_{\mathbf{x}} g(f_{\mathcal{W}'_L}(\mathbf{x}), y)\right\|_2 + \left\|\Phi^{-1}\nabla_{\mathbf{x}} g(f_{\mathcal{W}'_L}(\mathbf{x}), y) + (\Phi')^{-1}\nabla_{\mathbf{x}} g(f_{\mathcal{W}'_L}(\mathbf{x}), y)\right\|_2 \\
\leq&\left\|\Phi^{-1}\right\|_2\left\|\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}), y) - \nabla_{\mathbf{x}} g(f_{\mathcal{W}'_L}(\mathbf{x}), y)\right\|_2 + \left\|\Phi^{-1} - (\Phi')^{-1}\right\|_2\left\|\nabla_{\mathbf{x}} g(f_{\mathcal{W}'_L}(\mathbf{x}), y)\right\|_2 \\
\leq&\left(\left\|\Phi^{-1}\right\|_2 + \frac{1}{\kappa^2}\left\|\nabla_{\mathbf{x}} g(f_{\mathcal{W}'_L}(\mathbf{x}), y)\right\|_2\right)\left\|\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}), y) - \nabla_{\mathbf{x}} g(f_{\mathcal{W}'_L}(\mathbf{x}), y)\right\|_2 \qquad (4.13) \\
\leq&\frac{1}{\kappa}(1 + \frac{1}{\kappa}\prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}})\left\|\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}), y) - \nabla_{\mathbf{x}} g(f_{\mathcal{W}'_L}(\mathbf{x}), y)\right\|_2 \qquad (4.14) \\
\leq&\frac{1}{\kappa}(1 + \frac{1}{\kappa}\prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}})\prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}}\sum_{\mathrm{j}=1}^{L}\left(\frac{\tau_{\mathrm{j}}}{c_{\mathrm{j}}} + B\prod_{k=1}^{\mathrm{j}} c_k\sum_{k=1}^{\mathrm{j}}\frac{\tau_k}{c_k}\right). \qquad (4.15)
\end{aligned}
$$

Under the assumptions, Equation 4.13 is obtained by

$$
\begin{aligned}
\left\|\Phi^{-1} - (\Phi')^{-1}\right\|_2 =& \max_{t\in[p]}\frac{\left||\nabla_{\mathbf{x}}^{(t)} g(f_{\mathcal{W}'_L}(\mathbf{x}), y)| - |\nabla_{\mathbf{x}}^{(t)} g(f_{\mathcal{W}_L}(\mathbf{x}), y)|\right|}{|\nabla_{\mathbf{x}}^{(t)} g(f_{\mathcal{W}_L}(\mathbf{x}), y)||\nabla_{\mathbf{x}}^{(t)} g(f_{\mathcal{W}'_L}(\mathbf{x}), y)|} \\
\leq&\frac{1}{\kappa^2}\max_{t\in[p]}\left||\nabla_{\mathbf{x}}^{(t)} g(f_{\mathcal{W}'_L}(\mathbf{x}), y)| - |\nabla_{\mathbf{x}}^{(t)} g(f_{\mathcal{W}_L}(\mathbf{x}), y)|\right| \\
\leq&\frac{1}{\kappa^2}\max_{t\in[p]}|\nabla_{\mathbf{x}}^{(t)} g(f_{\mathcal{W}'_L}(\mathbf{x}), y) - \nabla_{\mathbf{x}}^{(t)} g(f_{\mathcal{W}_L}(\mathbf{x}), y)| \\
\leq&\frac{1}{\kappa^2}\left\|\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}), y) - \nabla_{\mathbf{x}} g(f_{\mathcal{W}'_L}(\mathbf{x}), y)\right\|_2.
\end{aligned}
$$

Equation 4.14 is obtained by $\left\|\Phi^{-1}\right\|_2 = \max_{t\in[p]}(1/\nabla_{\mathbf{x}}^{(t)} g(f_{\mathcal{W}_L}(\mathbf{x}), y)) \leq 1/\kappa$ and $\nabla_{\mathbf{x}} g(f_{\mathcal{W}'_L}(\mathbf{x}), y)) = \nabla_{\mathbf{x}} f_{\mathcal{W}'_L}(\mathbf{x})(\nabla g)(f_{\mathcal{W}'_L}(\mathbf{x}), y)) \leq \prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}}$. Equation 4.15 is obtained from Lemma 4.5.2. □

Now, we prove that under the assumptions of Lemma 4.5.2, given $\tau > 0$,

$$
\rho_m = \left(\frac{1}{m}\sum_{\mathrm{i}=1}^{m}\left\|f_{\mathcal{W}_L}(\mathbf{x}_{\mathrm{i}} + \epsilon\cdot\text{sign}(\nabla_{\mathbf{x}_{\mathrm{i}}} g(f_{\mathcal{W}_L}(\mathbf{x}_{\mathrm{i}}), y))) - f_{W'_L}(\mathbf{x}_{\mathrm{i}} + \epsilon\cdot\text{sign}(\nabla_{\mathbf{x}_{\mathrm{i}}} g(f_{\mathcal{W}'_L}(\mathbf{x}_{\mathrm{i}}), y)))\right\|_2^2\right)^{\frac{1}{2}} \leq \tau
$$

by choosing

$$
\tau_{\mathrm{j}} = \frac{\tau c_{\mathrm{j}}}{L(B + \sqrt{p}\epsilon + \Gamma)\prod_{\mathrm{i}=1}^{L} c_{\mathrm{i}}}, \qquad (4.16)
$$

where $\Gamma = \epsilon\frac{1}{\kappa}(1+\frac{1}{\kappa}\prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}})\prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}}(1+\frac{B}{L}\sum_{\mathrm{j}=1}^{L}(\mathrm{j}\prod_{k=1}^{\mathrm{j}} c_k))$.

First, we inductively prove that

$$\begin{aligned}
&\left\|f_{\mathcal{W}_L}(\mathbf{x}+\epsilon\cdot\operatorname{sign}(\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}),y))) - f_{W'_L}(\mathbf{x}+\epsilon\cdot\operatorname{sign}(\nabla_{\mathbf{x}} g(f_{\mathcal{W}'_L}(\mathbf{x}),y)))\right\|_2 \\
\leq&\prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}}\sum_{\mathrm{j}=1}^{L}\frac{\tau_{\mathrm{j}}}{c_{\mathrm{j}}}(B+\sqrt{p}\epsilon) + \prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}}\epsilon\left\|\operatorname{sign}(\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}),y)) - \operatorname{sign}(\nabla_{\mathbf{x}} g(f_{\mathcal{W}'_L}(\mathbf{x}),y))\right\|_2
\end{aligned}$$

When $L=0$, it is obvious that the above inequality holds. Then,

$$\begin{aligned}
&\left\|f_{\mathcal{W}_L}(\mathbf{x}+\epsilon\cdot\operatorname{sign}(\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}),y))) - f_{W'_L}(\mathbf{x}+\epsilon\cdot\operatorname{sign}(\nabla_{\mathbf{x}} g(f_{\mathcal{W}'_L}(\mathbf{x}),y)))\right\|_2 \\
\leq&\left\|W_L\sigma_{L-1}(f_{\mathcal{W}_{L-1}}(\mathbf{x}+\epsilon\cdot\operatorname{sign}(\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}),y)))) - W'_L\sigma_{L-1}(f_{\mathcal{W}_{L-1}}(\mathbf{x}+\epsilon\cdot\operatorname{sign}(\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}),y))))\right\|_2 \\
+&\left\|W'_L\sigma_{L-1}(f_{\mathcal{W}_{L-1}}(\mathbf{x}+\epsilon\cdot\operatorname{sign}(\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}),y)))) - W'_L\sigma_{L-1}(f_{\mathcal{W}'_{L-1}}(\mathbf{x}+\epsilon\cdot\operatorname{sign}(\nabla_{\mathbf{x}} g(f_{\mathcal{W}'_L}(\mathbf{x}),y))))\right\|_2 \\
\leq&\tau_L\prod_{\mathrm{j}=1}^{L-1} c_{\mathrm{j}}(B+\sqrt{p}\epsilon) + c_L\left\|f_{\mathcal{W}_{L-1}}(\mathbf{x}+\epsilon\cdot\operatorname{sign}(\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}),y))) - f_{\mathcal{W}'_{L-1}}(\mathbf{x}+\epsilon\cdot\operatorname{sign}(\nabla_{\mathbf{x}} g(f_{\mathcal{W}'_L}(\mathbf{x}),y)))\right\|_2 \\
\leq&\tau_L\prod_{\mathrm{j}=1}^{L-1} c_{\mathrm{j}}(B+\sqrt{p}\epsilon) + \prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}}\sum_{\mathrm{j}=1}^{L-1}\frac{\tau_{\mathrm{j}}}{c_{\mathrm{j}}}(B+\sqrt{p}\epsilon) + \prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}}\epsilon\left\|\operatorname{sign}(\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}),y)) - \operatorname{sign}(\nabla_{\mathbf{x}} g(f_{\mathcal{W}'_L}(\mathbf{x}),y))\right\|_2 \\
\leq&\prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}}\sum_{\mathrm{j}=1}^{L}\frac{\tau_{\mathrm{j}}}{c_{\mathrm{j}}}(B+\sqrt{p}\epsilon) + \prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}}\epsilon\left\|\operatorname{sign}(\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}),y)) - \operatorname{sign}(\nabla_{\mathbf{x}} g(f_{\mathcal{W}'_L}(\mathbf{x}),y))\right\|_2.
\end{aligned}$$

Applying Lemma 4.5.3 and Equation 4.16, we have

$$\begin{aligned}
&\left\|f_{\mathcal{W}_L}(\mathbf{x}+\epsilon\cdot\operatorname{sign}(\nabla_{\mathbf{x}} g(f_{\mathcal{W}_L}(\mathbf{x}),y))) - f_{W'_L}(\mathbf{x}+\epsilon\cdot\operatorname{sign}(\nabla_{\mathbf{x}} g(f_{\mathcal{W}'_L}(\mathbf{x}),y)))\right\|_2 \\
\leq&\prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}}\left(\sum_{\mathrm{j}=1}^{L}\frac{\tau_{\mathrm{j}}}{c_{\mathrm{j}}}(B+\sqrt{p}\epsilon) + \epsilon\frac{1}{\kappa}(1+\frac{1}{\kappa}\prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}})\prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}}\sum_{\mathrm{j}=1}^{L}\left(\frac{\tau_{\mathrm{j}}}{c_{\mathrm{j}}} + B\prod_{k=1}^{\mathrm{j}} c_k\sum_{k=1}^{\mathrm{j}}\frac{\tau_k}{c_k}\right)\right) \\
\leq&\tau
\end{aligned}$$

Therefore, the covering number of $\widetilde{\mathcal{F}}_{\mathcal{W}_L}$ is

$$\begin{aligned} N(\widetilde{\mathcal{F}}_{\mathcal{W}_L}, \tau, \rho_m) \leq & \prod_{\mathrm{j}=1}^{L} \sup_{\substack{W_1,\ldots,W_{\mathrm{j}-1} \\ \forall \mathrm{i}<\mathrm{j}, W_{\mathrm{i}} \in S_{c_{\mathrm{i}}, r_{\mathrm{i}}}}} N\left(\left\{W_{\mathrm{j}} : W_{\mathrm{j}} \in S_{c_{\mathrm{j}}, r_{\mathrm{j}}}\right\}, \tau_{\mathrm{j}}, \|\cdot\|_2\right) \\ \leq & \prod_{\mathrm{j}=1}^{L} \left(\frac{9c_{\mathrm{j}}}{\tau_{\mathrm{j}}}\right)^{r_{\mathrm{j}}(p_{\mathrm{j}}+p_{\mathrm{j}-1}+1)} \\ \leq & \left(\frac{9L(B+\sqrt{p}\epsilon+\Gamma)\prod_{\mathrm{j}=1}^{L} c_{\mathrm{j}}}{\tau}\right)^{(2p+1)\sum r_{\mathrm{j}}}. \end{aligned}$$

### 4.5.2 Proof of Lemma 4.2.2

*Proof.* We first scale $\mathbf{z} \in \mathcal{Z}$ to be a unit ball, denoting as $\mathcal{Z}/A$ for simplicity. For any $\tau_1 > 0$, there exists a $\tau_1$-covering of $\mathcal{Z}/A$ consisting of $N(\mathcal{Z}/A, \tau_1, \rho_\infty)$ balls: $\mathcal{B}_1, \ldots, \mathcal{B}_{N(\mathcal{Z}/A,\tau_1,\rho_\infty)}$. By [58], we have

$$N(\mathcal{Z}/A, \tau_1, \rho_\infty) = \left(\frac{3}{\tau_1}\right)^r$$

Choose the center $\mathbf{o}_t$ in each of the ball $\mathcal{B}_t$, for $t \in [N(\mathcal{Z}/A, \tau_1, \rho_\infty)]$. The function $\tilde{h}(\mathbf{z})$ on the set $\mathcal{Z}$ will be approximated by the construction:

$$\tilde{h}'(\mathbf{z}) = \left[\frac{2\tilde{h}(A\mathbf{o}_t)}{\tau_2}\right]\frac{\tau_2}{2} \quad \text{for } \mathbf{z} \in \mathcal{B}_t$$

Take $\tau_1 = \tau_2/(2MA)$, we have

$$\begin{aligned} \sup_{\mathbf{z}} |\tilde{h}(\mathbf{z}) - \tilde{h}'(\mathbf{z})| \leq & \sup_{\mathbf{z}} \left|\tilde{h}(\mathbf{z}) - \frac{2\tilde{h}(A\mathbf{o}_t)}{\tau_2}\frac{\tau_2}{2}\right| + \sup_{\mathbf{z}} \left|\frac{2\tilde{h}(A\mathbf{o}_t)}{\tau_2}\frac{\tau_2}{2} - \left[\frac{2\tilde{h}(A\mathbf{o}_t)}{\tau_2}\right]\frac{\tau_2}{2}\right| \\ \leq & \sup_{\mathbf{z}} |\tilde{h}(\mathbf{z}) - \tilde{h}(A\mathbf{o}_t)| + \frac{\tau_2}{2} \\ \leq & M \sup_{\mathbf{z}} \left\|\mathbf{z} - A\mathbf{o}_t\right\|_2 + \frac{\tau_2}{2} \\ \leq & MA \sup_{\mathbf{z}} \left\|\mathbf{z}/A - \mathbf{o}_t\right\|_2 + \frac{\tau_2}{2} \leq MA\tau_1 + \frac{\tau_2}{2} \leq \tau_2 \end{aligned}$$

Let $s = 2[4AM/\tau_2] + 1$. The function $\tilde{h}'(\mathbf{z})$ assumes no more than $N(\mathcal{Z}/A, \tau_1, \rho_\infty)$ values on each set $s$ and therefore, the total number of all functions is no greater than the number $s^{N(\mathcal{Z}/A,\tau_1,\rho_\infty)}$, that is,

$$N(\tilde{\mathcal{H}}, \tau_2, \rho_\infty) \leq \left(2\left\lceil\frac{4MA}{\tau_2}\right\rceil + 1\right)^{(\frac{6MA}{\tau_2})^r}.$$

□

### 4.5.3 Proof of Lemma 4.2.3

*Proof.* Note that Lemma 1 is also applied for any fixed $l \in [L]$. Using Lemma 1 and Lemma 2, we compute the Rademacher complexity for the decomposed DNN. First, we argue that for $\tau > 0$

$$N(g \circ \mathcal{H}_{r_l} \circ \tilde{\mathcal{F}}_{\mathcal{W}_l}, \tau, \rho_m) \leq N(g \circ \mathcal{H}_{r_l}, \frac{\tau}{2}, \rho_\infty) N(\tilde{\mathcal{F}}_{\mathcal{W}_l}, \frac{\tau}{2\prod_{\mathrm{j}=l}^{L} c_\mathrm{j}}, \rho_m).$$

Pick any function $\tilde{h}_{r_l} := g \circ h_{r_l} \in g \circ \mathcal{H}_{r_l}$ and $f_{\mathcal{W}_l} \in \tilde{\mathcal{F}}_{\mathcal{W}_l}$, and let $\tilde{h}'_{r_l}$ and $f'_{\mathcal{W}_l}$ be the closest function in $g \circ \mathcal{H}_{r_l}$ and $\tilde{\mathcal{F}}_{\mathcal{W}_l}$ respectively. Since $\tilde{h}'_{r_l}$ is $\prod_{\mathrm{j}=l}^{L} c_\mathrm{j}$-Lipschitz, we have

$$\begin{aligned}
\rho_m(\tilde{h}_{r_l} f_{\mathcal{W}_l}, \tilde{h}'_{r_l} f'_{\mathcal{W}_l}) =& \sqrt{\frac{1}{m}\sum_{\mathrm{i}=1}^{m} |\tilde{h}_{r_l} f_{\mathcal{W}_l} - \tilde{h}'_{r_l} f'_{\mathcal{W}_l}|^2} \\
\leq& \sqrt{\frac{1}{m}\sum_{\mathrm{i}=1}^{m} |\tilde{h}_{r_l} f_{\mathcal{W}_l} - \tilde{h}'_{r_l} f_{\mathcal{W}_l}|^2 + \frac{1}{m}\sum_{\mathrm{i}=1}^{m} |\tilde{h}'_{r_l} f_{\mathcal{W}_l} - \tilde{h}'_{r_l} f'_{\mathcal{W}_l}|^2} \\
\leq& \sup_{\tilde{\mathbf{x}}} |\tilde{h}_{r_l}(\tilde{\mathbf{x}}) - \tilde{h}'_{r_l}(\tilde{\mathbf{x}})| + \prod_{\mathrm{j}=l}^{L} c_\mathrm{j} \sqrt{\frac{1}{m}\sum_{\mathrm{i}=1}^{m} \|f_{\mathcal{W}_l} - f'_{\mathcal{W}_l}\|_2^2} \\
\leq& \frac{\tau}{2} + \frac{\tau}{2} = \tau
\end{aligned}$$

Therefore, we can choose $\tilde{h}'_{r_l}$ and $f'_{\mathcal{W}_l}$ from the covers of $g\circ\mathcal{H}_{r_l}$ and $\widetilde{\mathcal{F}}_{\mathcal{W}_l}$ to cover $g\circ\mathcal{H}_{r_l}\circ\widetilde{\mathcal{F}}_{\mathcal{W}_l}$. By standard Dudley's entropy integral, we have

$$\begin{aligned}
\widehat{\mathfrak{R}}_m(g\circ\mathcal{H}_{r_l}\circ\widetilde{\mathcal{F}}_{\mathcal{W}_l}) \lesssim& \inf_{\beta>0}\left\{\beta+\frac{1}{\sqrt{m}}\int_\beta^\alpha\sqrt{\ln N(g\circ\mathcal{H}_{r_l}\circ\widetilde{\mathcal{F}}_{\mathcal{W}_l},\tau,\rho_m)}d\tau\right\}\\
\leq& \inf_{\beta>0}\left\{\beta+\frac{1}{\sqrt{m}}\int_\beta^\alpha\sqrt{\ln N(g\circ\mathcal{H}_{r_l},\frac{\tau}{2},\rho_\infty)}d\tau+\frac{1}{\sqrt{m}}\int_\beta^\alpha\sqrt{\ln N(\widetilde{\mathcal{F}}_{\mathcal{W}_l},\frac{\tau}{2\prod_{\mathrm{j}=l}^L c_{\mathrm{j}}},\rho_m)}d\tau\right\}\\
\leq& \inf_{\beta>0}\left\{\beta+\frac{1}{\sqrt{m}}\int_\beta^\alpha\sqrt{\Big(\frac{12(B+\sqrt{p}\epsilon)\prod_{\mathrm{j}=1}^L c_{\mathrm{j}}}{\tau}\Big)^{r_l}\ln\Big(2\Big\lceil\frac{8(B+\sqrt{p}\epsilon)\prod_{\mathrm{j}=1}^L c_{\mathrm{j}}}{\tau}\Big\rceil+1\Big)}d\tau\right.\\
&\left.+\frac{1}{\sqrt{m}}\int_\beta^\alpha\sqrt{(2p+1)\sum_{\mathrm{j}=1}^l r_{\mathrm{j}}\ln\Big(\frac{18\prod_{\mathrm{j}=1}^L c_{\mathrm{j}}l(B+\sqrt{p}\epsilon+\Gamma_l)}{\tau}\Big)}d\tau\right\}\\
:=& \inf_{\beta>0}\{P+Q\},
\end{aligned}$$

where

$$\alpha=\sup_{\substack{\mathbf{x}\in\mathcal{X}\\ gh_{r_l}f_{\mathcal{W}_l}\in g\circ\mathcal{H}_{r_l}\circ\widetilde{\mathcal{F}}_{\mathcal{W}_l}}}\rho_m(gh_{r_l}f_{\mathcal{W}_l}(\mathbf{x}),0)=(B+\sqrt{p}\epsilon)\prod_{\mathrm{j}=1}^L c_{\mathrm{j}}.$$

We consider $Q$,

$$\begin{aligned}
Q=&\frac{1}{\sqrt{m}}\int_\beta^\alpha\sqrt{(2p+1)\sum_{\mathrm{j}=1}^l r_{\mathrm{j}}\ln\Big(\frac{18\prod_{\mathrm{j}=1}^L c_{\mathrm{j}}l(B+\sqrt{p}\epsilon+\Gamma_l)}{\tau}\Big)}d\tau\\
\leq&\frac{\alpha}{\sqrt{m}}\sqrt{(2p+1)\sum_{\mathrm{j}=1}^l r_{\mathrm{j}}\ln\Big(\frac{18l(\alpha+\prod_{\mathrm{j}=1}^l c_{\mathrm{j}}\Gamma_l)}{\beta}\Big)}
\end{aligned}$$

Then we consider $P$,

$$\begin{aligned}
P=&\beta+\frac{1}{\sqrt{m}}\int_\beta^\alpha\sqrt{\Big(\frac{12\alpha}{\tau}\Big)^{r_l}\ln\Big(2\Big\lceil\frac{8\alpha}{\tau}\Big\rceil+1\Big)}d\tau\\
\lesssim&\beta+\frac{1}{\sqrt{m}}\int_\beta^\alpha\sqrt{\Big(\frac{16\alpha}{\tau}\Big)^{r_l}\ln\Big(\frac{16\alpha}{\tau}\Big)}d\tau\\
\leq&\beta+\frac{32\alpha}{\sqrt{m}}\int_{-\sqrt{\ln\frac{16\alpha}{\beta}}}^{-\sqrt{\ln 16}}\mathrm{e}^{(r_l/2-1)t^2}t^2pt
\end{aligned}$$

(a) When $r_l/2-1<0$, i.e., $r_l=1$,

$$P \leq \beta + \frac{32\alpha}{\sqrt{m}} \int_{-\sqrt{\ln \frac{16\alpha}{\beta}}}^{-\sqrt{\ln 16}} \mathrm{e}^{(r_l/2-1)t^2} t^2 pt \leq \beta + \frac{16\alpha}{\sqrt{m}} \sqrt{\frac{2\pi}{2-r_l}} \mathbb{E}t^2 = \beta + \sqrt{\frac{2\pi}{m}} \frac{16\alpha}{(2-r_l)^{3/2}}$$

Therefore,

$$P+Q \leq \beta + \sqrt{\frac{2\pi}{m}} \frac{16\alpha}{(2-r_l)^{3/2}} + \frac{\alpha}{\sqrt{m}} \sqrt{(2p+1) \sum_{\mathrm{j}=1}^{l} r_{\mathrm{j}} \ln \left( \frac{18l(\alpha + \prod_{\mathrm{j}=1}^{l} c_{\mathrm{j}} \Gamma_l)}{\beta} \right)}$$

Take $\beta = \alpha/\sqrt{m}$, we have

$$\widehat{\mathfrak{R}}_m(g \circ \mathcal{H}_{r_l} \circ \widetilde{\mathcal{F}}_{\mathcal{W}_l}) \lesssim \frac{\alpha}{\sqrt{m}} \left( 1 + 16\sqrt{2\pi} + \sqrt{(2p+1) \sum_{\mathrm{j}=1}^{l} r_{\mathrm{j}} \ln \left( 18l\sqrt{m}(1 + \frac{\Gamma_l}{B+\sqrt{p}\epsilon}) \right)} \right)$$

(b) When $r_l/2-1=0$, i.e., $r_l=2$,

$$P \leq \beta + \frac{32\alpha}{\sqrt{m}} \int_{-\sqrt{\ln \frac{16\alpha}{\beta}}}^{-\sqrt{\ln 16}} t^2 pt = \beta + \frac{32\alpha}{3\sqrt{m}} \left( \left( \ln \frac{16\alpha}{\beta} \right)^{3/2} - (\ln 16)^{3/2} \right)$$

Hence,

$$P+Q \leq \beta + \frac{32\alpha}{3\sqrt{m}} \left( \left( \ln \frac{16\alpha}{\beta} \right)^{3/2} - (\ln 16)^{3/2} \right) + \frac{\alpha}{\sqrt{m}} \sqrt{(2p+1) \sum_{\mathrm{j}=1}^{l} r_{\mathrm{j}} \ln \left( \frac{18l(\alpha + \prod_{\mathrm{j}=1}^{l} c_{\mathrm{j}} \Gamma_l)}{\beta} \right)}$$

Choose $\beta = \alpha/\sqrt{m}$, we have

$$\widehat{\mathfrak{R}}_m(g \circ \mathcal{H}_{r_l} \circ \widetilde{\mathcal{F}}_{\mathcal{W}_l}) \lesssim \frac{\alpha}{\sqrt{m}} \left( 1 + 16 \left( \left( \ln(16\sqrt{m}) \right)^{3/2} - (\ln 16)^{3/2} \right) + \sqrt{(2p+1) \sum_{\mathrm{j}=1}^{l} r_{\mathrm{j}} \ln \left( 18l\sqrt{m}(1 + \frac{\Gamma_l}{B+\sqrt{p}\epsilon}) \right)} \right)$$

(c) When $r_l/2-1>0$, i.e., $r_l>2$,

$$P \leq \beta + \frac{32\alpha}{\sqrt{m}} \int_{-\sqrt{\ln \frac{16\alpha}{\beta}}}^{-\sqrt{\ln 16}} \mathrm{e}^{(r_l/2-1)t^2} t^2 pt \leq \beta + \frac{16\alpha}{r_l/2-1} \sqrt{\frac{\ln(16\alpha/\beta)}{m}} \left( \left( \frac{16\alpha}{\beta} \right)^{r_l/2-1} - 16^{r_l/2-1} \right)$$

Therefore,

$$P+Q \leq \beta+\frac{16\alpha}{r_l/2-1}\sqrt{\frac{\ln(16\alpha/\beta)}{m}}\left(\left(\frac{16\alpha}{\beta}\right)^{r_l/2-1}-16^{r_l/2-1}\right)+\frac{\alpha}{\sqrt{m}}\sqrt{(2p+1)\sum_{\mathrm{j}=1}^{l} r_{\mathrm{j}} \ln\left(\frac{18l(\alpha+\prod_{\mathrm{j}=1}^{l} c_{\mathrm{j}}\Gamma_l)}{\beta}\right)}$$

Let $\beta = \alpha/\sqrt[r_l]{m}$, we have

$$\widehat{\mathfrak{R}}_m(g\circ\mathcal{H}_{r_l}\circ\widetilde{\mathcal{F}}_{\mathcal{W}_l}) \lesssim \frac{\alpha}{\sqrt[r_l]{m}}\left(1+\frac{32}{r_l-2}\sqrt{16^{r_l-2}\ln(16\sqrt[r_l]{m})}+\sqrt{(2p+1)\sum_{\mathrm{j}=1}^{l} r_{\mathrm{j}} \ln\left(18l\sqrt[r_l]{m}(1+\frac{\Gamma_l}{B+\sqrt{p}\epsilon})\right)}\right).$$

This completes the proof. □

# 5. ON THE LATENT SPACE OF GENERATIVE MODELS

In chapter 1, we briefly depict the harm of dimensional mismatch between the latent distribution and data distribution, such as mode collapsing and wrong representation learning. To handle these drawbacks, we further explore the reason behind the mismatch phenomena, and propose a novel approach, called Latent Wasserstein GAN (LWGAN), to identify the intrinsic dimension of a data distribution that lies on a continuous manifold. In particular, we take the advantages of WAE and WGAN to learn an informative prior distribution $P_Z$ rather than using a fixed distribution such as standard Gaussian in the conventional methods. Our main contributions are summarized below:

- By modifying the latent distribution, we propose a new framework called LWGAN that combines the WGAN and WAE to adaptively learn the intrinsic dimension of a data distribution $P_X$.
- We theoretically establish the existence of a generator $G$ and an encoder $Q$ such that the intrinsic dimension of the encodes distribution $P_{Q(X)}$ is equal to the intrinsic dimension of $P_X$, and the generated data by these encodes follows the distribution $P_X$.
- We provide theoretical verification that our estimated intrinsic dimension is consistent with the true intrinsic dimension, and establish an upper bound to the generalization error of the LWGAN.
- We experimentally confirm that the LWGAN is able to detect the correct intrinsic dimension under several settings using both simulated examples as well as real datasets such as MNIST and CelebA. Meanwhile, LWGAN can generate high-quality synthetic data by the latent variable from our learned latent distribution.

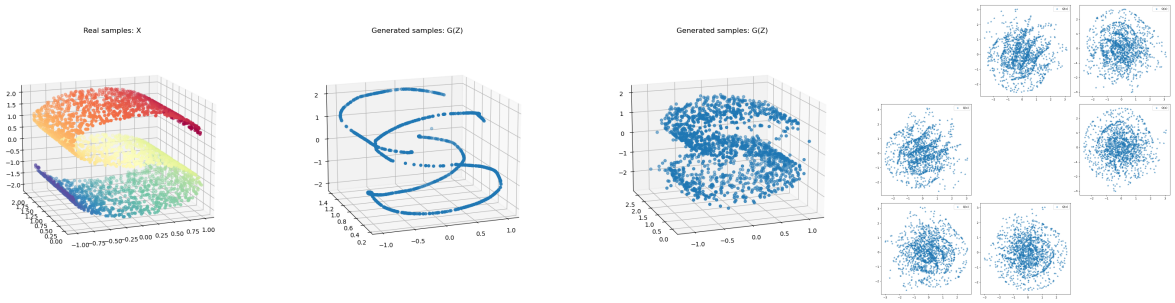
## 5.1 Latent Dimension Mismatch and the Encoder

Assume the data sample $\mathbf{x} \sim P_X$ to be a $p$-dimensional vector in $\mathbb{R}^p$, whose distribution $P_X$ is supported on a $r$-dimensional manifold $\mathcal{X}$. Consequently, define the intrinsic dimension of data distribution as the dimension of the manifold $\mathcal{X}$, denoted by $\text{InDim}(P_X)$, and its am-

bient dimension as the dimension of the ambient Euclidean space, denoted by $\mathrm{AmDim}(P_X)$. Here $\mathrm{InDim}(P_X) = r$, $\mathrm{AmDim}(P_X) = p$, and it is obvious that $\mathrm{InDim}(P_X)$ cannot be larger than $\mathrm{AmDim}(P_X)$. In terms of the latent prior distribution $P_Z$ with domain $\mathcal{Z}$, it is usually selected as a $d$-dimensional standard normal distribution $N(0, I_d)$ in most existing generative models, so $\mathrm{InDim}(P_Z) = \mathrm{AmDim}(P_Z) = d$. The dimension $d$ is typically predetermined to be a number that is smaller than $p$, thus $\mathrm{InDim}(P_Z)$ may not equal to $\mathrm{InDim}(P_X)$. In GAN-based models, if the generator $G$ is a continuous function then the synthetic sample $G(Z)$ mapped from the latent space will be supported on a manifold of dimension at most $\mathrm{InDim}(P_Z)$. When $\mathrm{InDim}(P_Z) < \mathrm{InDim}(P_X)$, pushing $P_{G(Z)}$ with unmatched intrinsic dimension to close to $P_X$ is a challenging task. On the other hand, in encoder-based models, the same phenomena of mismatch occurs for the encoded distribution $P_{Q(X)}$ obtained by the continuous encoder $Q$. In other words, it is difficult to enforce $P_{Q(X)}$ to be similar to $P_Z$ if $\mathrm{InDim}(P_X) < \mathrm{InDim}(P_Z)$ as filling a plane with a one dimensional curve is difficult.

Towards investigating the cause, we employ a toy example to provide intuition for the effects and consequences resulting from model and data distributions that possess differing intrinsic dimension. We consider a 3D S-Curve dataset as shows in Figure 5.1a, where each sample $\mathbf{x} = (x_1, x_2, x_3) \in \mathbb{R}^3$ is a vector with $x_1 = \sin(3\pi(\mathrm{i} - 0.5))$, $x_2 = 2\mathrm{j}$, $x_3 = \mathrm{sign}(3\pi(\mathrm{i} - 0.5))\cos(3\pi(\mathrm{i} - 0.5))$, for $\mathrm{i} \sim U[0, 1)$ and $\mathrm{j} \sim N(0, 1)$. This example results in $\mathrm{AmDim}(P_X) = 3$ and $\mathrm{InDim}(P_X) = 2$. If we choose the latent distribution $P_Z$ to be a 1-dimensional normal distribution $N(0, 1)$, the generative samples from the WGAN are plotted in Figure 5.1b. To minimize the 1-Wasserstein distance between the real distribution $P_X$ and generated distribution $P_{G(Z)}$, the WGAN can only learn an outer contour of the S-Curve but cannot fill points on the surface. Instead, we choose a 3-dimensional standard normal $N(0, I_3)$ as the latent distribution and train the data by the WAE. The WAE is forced to reconstruct the images well, while at the same time trying to fill the latent space evenly as a normal distribution with the 2-dimensional data manifold. The only way to do this is by curling the manifold up in the latent space as shows in Figure 5.1d. This disparity between $P_Z$ and $P_{Q(X)}$ in the latent space induces a poor generation of $P_{G(Z)}$ in Figure 5.1c.

A natural solution to this mismatch problem is to select a prior distribution $P_Z$ whose intrinsic dimension is the same as that of the data distribution. However, $\mathrm{InDim}(P_X)$ is




(a) S-Curve (b) WGAN: Generation (c) WAE: Generation (d) WAE: Latent Space

**Figure 5.1.** Illustrations of data generation with wrong dimensional latent space of WGAN and WAE. (a) Real data of S-Curve from $P_X$; (b) Generative samples by WGAN trained with a 1-dimensional standard normal distribution $P_Z$; (c) Generative samples by WAE trained with 3-dimensional standard normal distribution $P_Z$; (d) The ith component of $Q(X)$ against the jth component of $Q(X)$ of the learned latent distribution $P_{Q(X)}$ by WAE.

unknown explicitly, so one option involves to instead learn it from the data distribution. When both the continuous generator $G$ and the continuous encoder $Q$ are combined in an encoder generative model, $P_{G(Z)} = P_X$ and $P_{Q(X)} = P_Z$ cannot be reached simultaneously unless $\text{InDim}(P_X) = \text{InDim}(P_Z)$ according to our previous discussion. This motivates us to search for an encoder $Q$ such that $Q(X)$ reflects the latent space supported on a $r$-dimensional manifold, and a corresponding generator $G$ such that generated samples using these latent encodes are high-quality. To be concrete, we need an auto-encoder generative model that satisfies the following four goals at the same time: (a) The prior distribution $P_Z$ is supported on a $r$-dimensional manifold; (b) The encodes $Q(X)$ has a similar distribution with $Z$; (c) The distribution of $G(Z)$ is similar to $P_X$; (d) The difference between $X$ and $G(Q(X))$ is small.

Unlike those conventional generative models applying a fixed standard normal distribution to be the latent distribution, we consider a latent prior distribution whose intrinsic dimension could be less than $d$ to achieve the first goal. This idea is realized by the generalized definition of the normal distribution. In particular, let $Z_0$ be a standard multivariate

normal distribution from $N(0, I_d)$, then for a $d \times d$ lower triangular matrix $A$, $Z = AZ_0$ is also a normal distribution with the form

$$P_Z = N(0, AA^T),$$

where $AA^T$ is constrained to be a positive semi-definite covariance. When $A$ is a full-rank matrix, $P_Z$ is a multivariate normal distribution on $\mathbb{R}^d$. When $\text{rank}(A) < d$, some elements of $Z$ can be represented as the linear combination of other elements, so $P_Z$ degenerates to a normal distribution supported on $\text{rank}(A)$-dimensional subspace, and its corresponding intrinsic dimension becomes $\text{rank}(A)$. If $\text{rank}(A) = r$, the latent variable $Z$ can be mapped to $G(Z)$ supported on a $r$-dimensional manifold, meanwhile, the intrinsic dimensions of $P_Z$ and the encodes distribution $P_{Q(X)}$ can be the same. This consequently solves the mismatch phenomenon. The following Theorem Theorem 5.1.1 and Corollary 5.1.1 confirm that for any distribution residing on a smooth Riemannian manifold, there always exist an encoder $Q^* : \mathcal{X} \to \mathcal{Z}$ which guarantees meaningful encodings on $r$-dimensional manifold, and a generator $G^* : \mathcal{Z} \to \mathcal{X}$ which generates samples with the same distribution as data points by using these meaningful codes.

According to the Whitney embedding theorem [59], [60], every $r$-dimensional smooth manifold admits a smooth embedding into the Euclidean space $\mathbb{R}^{2r}$. Let us denote this embedding as $u : \mathcal{X} \to \mathbb{R}^{2r}$, and its image embedded in $\mathbb{R}^{2r}$ as $\mathcal{S} = u(\mathcal{X})$. Then the manifold $\mathcal{X}$ is diffeomorphic to image $\mathcal{S}$. Different property of $\mathcal{S}$ will lead to different conclusions as follows.

**Theorem 5.1.1.** *Consider a continuous random variable $X$ from the distribution $P_X$ supported on a $r$-dimensional smooth manifold $\mathcal{X}$. Denote $\mathcal{S} = u(\mathcal{X})$ as an embedded submanifold in $\mathbb{R}^{2r}$, where $u : \mathcal{X} \to \mathbb{R}^{2r}$ is a smooth function. Assume that there exists a continuous function $h$, such that $\mathcal{S}$ can be represented as a graph of this continuous function, i.e.,*

$$\mathcal{S} = \{(\tilde{X}^{(1)}, \tilde{X}^{(2)}) \in \mathbb{R}^r \times \mathbb{R}^r : \tilde{X}^{(1)} \in V \subseteq \mathbb{R}^r \text{ and } \tilde{X}^{(2)} = h(\tilde{X}^{(1)})\},$$


(a) Bijective (b) Surjective

**Figure 5.2.** (a) The transformations $\mathcal{X} \to \mathcal{Z}$ and its inverse $\mathcal{Z} \to \mathcal{X}$ in Theorem 5.1.1 are both deterministic. (b) In Corollary 5.1.1, the transformation $\mathcal{X} \to \mathcal{Z}$ is deterministic, while its reverse $\mathcal{Z} \to \mathcal{X}$ is stochastic.

*where $V \subseteq \mathbb{R}^r$ is an open set. Then there exist a $d$-dimensional degenerated multivariate normal distribution $N(0, A^*A^{*T})$ supported on $r$-dimensional manifold $\mathcal{Z}$ with* $\operatorname{rank}(A^*) = r$*, and two mappings $Q^* : \mathcal{X} \to \mathcal{Z}$ and $G^* : \mathcal{Z} \to \mathcal{X}$, such that $Q^*(X) \sim N(0, A^*A^{*T})$ and $G^* \circ Q^*$ is an identity mapping, i.e., $X = G^*(Q^*(X))$.*

Theorem Theorem 5.1.1 establishes the existence of both the encoder $Q^*$ and the generator $G^*$ under the circumstance when $\mathcal{S}$ is a continuous function graph. Theorem Theorem 5.1.1 also indicates that $Q^*$ is at least an injective mapping, hence we can regard $G^*$ as its left inverse. Moreover, since $P_{Q^*(X)} = P_{A^*Z_0}$ and $X = G^*(Q^*(X))$, it is obvious that $P_X = P_{G^*(A^*Z_0)}$. On the other hand, when $\mathcal{S}$ does not possess the property of a continuous function graph, we provide a more general conclusion with the help of the Noise-Outsourcing Lemma [61], [62].

**Lemma 5.1.1** (Noise-Outsourcing Lemma)**.** *Let $(X, Z)$ be a random pair taking values in $\mathcal{X} \times \mathcal{Z}$ with joint distribution $P_{X,Z}$. Suppose $\mathcal{X}$ and $\mathcal{Z}$ are standard Borel spaces. Then there exist a random variable $\eta \sim P_\eta$ and a Borel-measurable function $G : \mathbb{R} \times \mathcal{Z} \to \mathcal{X}$ such that $\eta$ is independent of $\mathcal{Z}$ and*

$$(X, Z) = (G(\eta, Z), Z) \text{ almost surely.}$$

The noise-outsourcing lemma provides a unified view of distribution estimation. If the joint distribution of $(G(\eta, Z), Z)$ is the same as that of $(X, Z)$, it is equivalent to matching

the marginal distribution $P_X$ with $G(\eta, Z)$ when the same marginal distribution of $Z$ is involved. Typically $P_\eta$ is a simple distribution such as uniform distribution $U[0,1)$ and standard Gaussian $N(0,1)$.

**Corollary 5.1.1.** *Consider a continuous random variable $X$ from the distribution $P_X$ supported on a $r$-dimensional smooth manifold $\mathcal{X}$. Then there exist a $d$-dimensional degenerated multivariate normal distribution $N(0, A^*A^{*T})$ supported on $r$-dimensional manifold $\mathcal{Z}$ with $\text{rank}(A^*) = r$, a mapping $Q^* : \mathcal{X} \to \mathcal{Z}$ such that $Q^*(X) \sim N(0, A^*A^{*T})$, and a stochastic transformation $G^* : \mathbb{R} \times \mathcal{Z} \to \mathcal{X}$ with a random variable $\eta \sim P_\eta$ such that $G^*(\eta, Q^*(X))$ follows the same distribution as $X$.*

The $Q^*$ constructed in the Corollary 5.1.1 could be a subjective function, but it is not invertible since multiple inputs can map to the same output. Hence we need to construct a stochastic inverse $G^*$ that transforms the code $\mathbf{z} \in \mathcal{Z}$ to the data distribution $P_X$ as illustrates in Figure 5.2b. Corollary 5.1.1 presents that $Q^*(X)$ has the same distribution as $Z \sim N(0, A^*A^{*T})$, thus with the help of Lemma 5.1.1, we are able to claim that $G^*(\eta, Q^*(X))$ follows the same distribution as $X$.

So far, Theorem Theorem 5.1.1 and Corollary 5.1.1 provide us a feasible way to identify the dimension of the data manifold $\mathcal{X}$ by learning a latent distribution with the same intrinsic dimension via the encoder $Q$. Since $\text{InDim}(P_Z) = \text{InDim}(P_{AZ_0}) = \text{rank}(AA^T)$, in practice, we can compute $\text{rank}(AA^T)$ by counting the number of non-zero eigenvalues of the matrix $AA^T$ through the eigenvalue decomposition. This allows for identification of the intrinsic dimension of the data manifold $\mathcal{X}$ as $\text{InDim}(P_X) = \text{rank}(A^*A^{*T}) = r$.

## 5.2 Latent Wasserstein GAN

We are ready to take advantages of both the WGAN and the WAE to formulate our new auto-encoder generative model, called LWGAN, which is capable of learning $A^*$, $Q^*$, and $G^*$ that simultaneously accomplish our four goals. We will mainly focus on the case in Theorem 1 where $\mathcal{S}$ is assumed to be a continuous graph. The algorithm for the general case in Corollary 1 is similar and we provide the discussion in chapter 6.

We start from the primal and dual format of Wasserstein distance. Let the latent variable $Z \in \mathbb{R}^d$ be from a normal distribution whose covariance is $AA^T$ where $A$ is a rank-$r$ matrix. Hence, $Z = AZ_0$ where $Z_0 \in \mathbb{R}^d$ is a standard multivariate normal vector. Recall that the primal format of 1-Wasserstein distance between data distribution $P_X$ and generative distribution $P_{G(Z)}$ is

$$W_1(P_X, P_{G(Z)}) = \inf_{\pi \in \Pi(P_X, P_Z)} \mathbb{E}_{(X,Z)\sim\pi} \|X - G(Z)\|, \tag{5.1}$$

and its beautiful dual format is

$$W_1(P_X, P_{G(Z)}) = \sup_{f \in \mathcal{F}} \Big\{ \mathbb{E}_X f(X) - \mathbb{E}_Z f(G(Z)) \Big\}. \tag{5.2}$$

Both the primal Wasserstein distance Equation 5.1 and dual Wasserstein distance Equation 5.2 are constrained optimization problems. For the primal problem, two constraints are that the marginal distributions of $\boldsymbol{\pi}(\mathbf{x}, \mathbf{z})$ are equivalent to $P_X$ and $P_Z$ respectively. Since the primal variable $f$ in the dual problem Equation 5.2 is also a dual variable for the primal problem Equation 5.1, the optimal value of the primal problem using the Lagrange multipliers is

$$\begin{aligned}
&\inf_{\pi} \mathbb{E}_{\pi} \Big\| X - G(Z) \Big\| + \int_{\mathbf{x}} f(\mathbf{x}) \Big( p_X(\mathbf{x}) - \int_{\mathbf{z}} \boldsymbol{\pi}(\mathbf{x}, \mathbf{z}) d\mathbf{z} \Big) d\mathbf{x} - \int_{\mathbf{z}} f(G(\mathbf{z})) \Big( p_Z(\mathbf{z}) - \int_{\mathbf{x}} \boldsymbol{\pi}(\mathbf{x}, \mathbf{z}) d\mathbf{x} \Big) d\mathbf{z} \\
&= \inf_{Q \in \mathcal{Q}} \mathbb{E}_X \Big[ \|X - G(Q(X))\| + f(G(Q(X))) \Big] - \mathbb{E}_{Z_0} \Big[ f(G(AZ_0)) \Big],
\end{aligned} \tag{5.3}$$

where an encoder $Q$ is introduced to approximate the conditional distribution of $Z$ given $X$, and two Lagrange multipliers are $f(\mathbf{x})$ and $-f(G(\mathbf{z}))$ respectively. On the other hand, the constraint for the dual problem is that $f$ needs to be 1-Lipschitz for the input $f(\mathbf{x}) - f(G(\mathbf{z})) \leq \|\mathbf{x} - G(\mathbf{z})\|$. Similarly, we can write the optimal value of the dual problem as

$$\begin{aligned}
&\sup_{f \in \mathcal{F}} \mathbb{E}_X \Big\{ f(X) \Big\} - \mathbb{E}_Z \Big\{ f(G(Z)) \Big\} - \int_{\mathcal{X} \times \mathcal{Z}} \boldsymbol{\pi}(\mathbf{x}, \mathbf{z}) \Big( f(\mathbf{x}) - f(G(\mathbf{z})) - \|\mathbf{x} - G(\mathbf{z})\| \Big) d\mathbf{x} d\mathbf{z} \\
=& \sup_{f \in \mathcal{F}} \Big\{ \mathbb{E}_X \Big[ \|X - G(Q(X))\| + f(G(Q(X))) \Big] - \mathbb{E}_{Z_0} \Big[ f(G(AZ_0)) \Big] \Big\},
\end{aligned} \tag{5.4}$$

where we use the Lagrange multiplier $\boldsymbol{\pi}(\mathbf{x}, \mathbf{z})$ for the 1-Lipschitz constraint.

Corresponding to the iterative update between the minimization problem Equation 5.3 and maximization problem Equation 5.4, we define a novel distance between the real data distribution $P_X$ and generated data distribution $P_{G(AZ_0)}$ given a generator $G$ as

$$\overline{W}_1(P_X, P_{G(AZ_0)}) = \inf_{Q\in\mathcal{Q}} \sup_{f\in\mathcal{F}} \Big\{\mathbb{E}_X\|X - G(Q(X))\| + \mathbb{E}_X\Big[f(G(Q(X)))\Big] - \mathbb{E}_{Z_0}\Big[f(G(AZ_0))\Big]\Big\}. \tag{5.5}$$

Here $\mathcal{F}$ is a set of all bounded 1-Lipschitz functions, and $\mathcal{Q}$ is a set of encoder mappings. The term $\mathbb{E}_X\|X - G(Q(X))\|$ can be treated as the autoencoder reconstruction error in the WAE as well as a loss to match the distributions between $X$ and $G(Q(X))$. Another term $\mathbb{E}_X[f(G(Q(X)))] - \mathbb{E}_{Z_0}[f(G(AZ_0))]$ can be treated as a loss for the generator as well as a loss to match the distribution between $G(Q(X))$ and $G(AZ_0)$. We emphasize that this term is different with the objective function of the WGAN in Equation 5.2. Based on this definition, finding the $\overline{W}_1$ distance between $P_X$ and $P_{G(AZ_0)}$ exactly matches solving the primal problem Equation 5.3 and dual problem Equation 5.4 iteratively. Obviously, Equation 5.5 reaches its minimum as the 1-Wasserstein distance $W_1(P_X, P_{G(AZ_0)})$ when $P_{Q^*(X)} = P_{AZ_0}$, which is illustrated by the following theorem.

**Theorem 5.2.1.** *The $\overline{W}_1$ distance has the following property*

$$\overline{W}_1(P_X, P_{G(AZ_0)}) = \inf_{Q\in\mathcal{Q}} \Big\{W_1(P_X, P_{G(Q(X))}) + W_1(P_{G(Q(X))}, P_{G(AZ_0)})\Big\}. \tag{5.6}$$

*Therefore, $W_1(P_X, P_{G(AZ_0)}) \leq \overline{W}_1(P_X, P_{G(AZ_0)})$, and the equality hold if there exists a $Q^* \in \mathcal{Q}$ such that $Q^*(X)$ has the same distribution with $AZ_0$.*

Finally, we obtain the generator $G$ and matrix $A$ by minimizing the new distance $\overline{W}_1$:

$$\min_{G\in\mathcal{G}, A\in\mathcal{A}} \overline{W}_1(P_X, P_{G(AZ_0)}), \tag{5.7}$$

where $\mathcal{G}$ is a set of generator mappings, and $\mathcal{A} \subseteq \mathbb{R}^{d\times d}$ is a set of low-rank matrices. As a result, our goals that $P_{Q^*(X)} = P_{A^*Z_0}$ and $P_X = P_{G^*(A^*Z_0)}$ are attained when Equation 5.7 achieves the optimal solution. In practice, we minimize the empirical version of $\overline{W}_1$ by

replacing the expectation by the Monte Carlo average. The mappings $Q$, $G$ and $f$ are parametrised with deep neural nets by parameters $\theta_Q$, $\theta_G$ and $\theta_f$ respectively, in which case back propagation can be used with stochastic gradient descent techniques to optimize the objective. We assume that these network spaces are large enough to include the true rank-$r$ matrix $A^*$, encoder $Q^*$, generator $G^*$, and the optimal discriminator $f^*$ such that Equation 5.7 achieve the minimum value. This is not a strong assumption due to the universal approximation theorem of DNNs [63]. Furthermore, some regularization terms are added in order to push $Q(X)$ to $Z$, and guarantee $A$ to be a low rank matrix. Thus, LWGAN is a minimax optimization problem solving

$$\min_{\theta_G,\theta_Q,A}\max_{\theta_f}\left\{\frac{1}{m}\sum_{\mathrm{i}=1}^{m}\ell(\mathbf{x}_\mathrm{i},\mathbf{z}_\mathrm{i};\theta_G,\theta_Q,A,\theta_f)-\lambda_1 J_1(\theta_f)+\lambda_2 J_2(\theta_Q,A)+\lambda_3 J_3(A),\right\} \tag{5.8}$$

where

$$\ell(\mathbf{x},\mathbf{z};\theta_G,\theta_Q,A,\theta_f)=\|\mathbf{x}-G(Q(\mathbf{x};\theta_Q);\theta_G)\|+f(G(Q(\mathbf{x};\theta_Q);\theta_G);\theta_f)-f(G(A\mathbf{z}_0;\theta_G);\theta_f). \tag{5.9}$$

Since $f$ is assumed to be 1-Lipschitz, we adopt the gradient penalty defined as $J_1(\theta_f)=\mathbb{E}_X\Big\{(\|\nabla_X f(X;\theta_f)\|_2-1)^2\Big\}$ in [36] to enforce the 1-Lipschitz constraint on $f\in\mathcal{F}$. We use the MMD penalty [64], denoted by $J_2(\theta_Q,A)=\mathrm{MMD}_\kappa(P_{Q(X;\theta_Q)},P_{AZ_0})$, to enforce $Q(X)$ to converge to $P_{AZ_0}$. The exact form of $J_2(\theta_Q,A)$ is

$$J_2(\theta_Q,A)=\frac{1}{m(m-1)}\sum_{\mathrm{i}\neq\mathrm{j}}\kappa(A\mathbf{z}_{0,\mathrm{i}},A\mathbf{z}_{0,\mathrm{j}})+\frac{1}{m(m-1)}\sum_{\mathrm{i}\neq\mathrm{j}}\kappa(Q(\mathbf{x}_\mathrm{i}),Q(\mathbf{x}_\mathrm{j}))-\frac{2}{m^2}\sum_{\mathrm{i},\mathrm{j}}\kappa(A\mathbf{z}_{0,\mathrm{i}},Q(\mathbf{x}_\mathrm{j})),$$

where $\kappa$ is set to be the Gaussian radial kernel function $\kappa(x,y)=\exp(\frac{-\|x-y\|^2}{2})$. Additionally, since $A$ needs to be a low rank matrix, the nuclear norm defied as the sum of singular value $\|A\|_*=\sum_{l=1}^{d}\sigma_l(A)$, which is a convex envelope of the rank function $\mathrm{rank}(A)$, is used as $J_3(A)$. We initialize $A$ as an identity matrix. We pre-specify some values for $\lambda_1$, $\lambda_2$ and $\lambda_3$, then the optimal tuning parameters are selected by grid search using cross validation. What is more, our new defined distance $\overline{W}_1$ provides the following duality gap as a natural measure

**Algorithm 1** The training algorithm of LWGAN

**Require:** The regularization coefficients $\lambda_1$, $\lambda_2$, and $\lambda_3$, tolerance for loss $\epsilon_1$ and DualGap $\epsilon_2$, and running steps $T$

Initialization $(\theta_G^0, \theta_Q^0, \theta_f^0, A^0)$
**while** $\frac{1}{m}\sum_{\mathrm{i}=1}^{m} \ell(\mathbf{x}_\mathrm{i}, \mathbf{z}_\mathrm{i}; \theta_G^k, \theta_Q^k, A^k, \theta_f^k) > \epsilon_1$ or $\mathrm{DualGap}(\theta_G^k, \theta_Q^k, A^k, \theta_f^k) > \epsilon_2$ **do**
  **for** $t = 1, ..., T$ **do**
    Sample real data $\{\mathbf{x}_\mathrm{i}^k\}_{\mathrm{i}=1}^m \sim P_X$, latent data $\{\mathbf{z}_{0,\mathrm{i}}^k\}_{\mathrm{i}=1}^m \sim N(0, I_d)$ and $\{\epsilon_\mathrm{i}\}_{\mathrm{i}=1}^m \sim U[0,1]$
    Set $\hat{\mathbf{x}}_\mathrm{i}^k \leftarrow \epsilon_\mathrm{i}\mathbf{x}_\mathrm{i}^k + (1-\epsilon_\mathrm{i})G(A^k\mathbf{z}_{0,\mathrm{i}}^k; \theta_G^k)$, $\mathrm{i} = 1, ..., m$ for the calculation of gradient penalty
    Calculate: $\hat{R}^k = \frac{1}{m}\sum_{\mathrm{i}=1}^{m} \ell(\mathbf{x}_\mathrm{i}^k, \mathbf{z}_\mathrm{i}^k; \theta_G^k, \theta_Q^k, A^k, \theta_f^k)$, $J_1(\theta_f^k) = (\|\nabla_{\hat{\mathbf{x}}^k} f(\hat{\mathbf{x}}^k; \theta_f^k)\|_2 - 1)^2$, and $-\nabla_f(\hat{R}_m^k + J_1(\theta_f^k))$
    Update $\theta_f$ by Adam: $\theta_f^{k+1} \leftarrow \theta_f^k + Adam(-\nabla_f(\hat{R}^k + J_1(\theta_f^k)))$
  **end for**
  **for** $t = 1, ..., T$ **do**
    Sample real data $\{\mathbf{x}_\mathrm{i}^k\}_{\mathrm{i}=1}^m \sim P_X$, latent variable $\{\mathbf{z}_{0,\mathrm{i}}^k\}_{\mathrm{i}=1}^m \sim N(0, I_d)$
    Calculate: $\hat{R}'^k = \frac{1}{m}\sum_{\mathrm{i}=1}^{m} \ell(\mathbf{x}_\mathrm{i}^k, \mathbf{z}_\mathrm{i}^k; \theta_G^k, \theta_Q^k, A^k, \theta_f^{k+1})$, $J_2(\theta_Q^k, A^k)$, $J_3(A^k)$ and $\nabla_{G,Q,A}(\hat{R}'^k + J_2(\theta_Q^k, A^k) + J_3(A^k))$
    Update $\theta_G$, $\theta_Q$, $A$ by Adam: $(\theta_G^{k+1}, \theta_Q^{k+1}, A^{\mathrm{i}+1}) \leftarrow (\theta_G^k, \theta_Q^k, A^k) + Adam(\nabla_{G,Q,A}(\hat{R}'^k + J_2(\theta_Q^k, A^k) + J_3(A^k)))$
  **end for**
  $k \leftarrow k + 1$
**end while**

to the convergence of the optimization Equation 5.8. For a given tuple $(\theta_G, \theta_Q, A, \theta_f)$, the duality gap is defined as

$$\mathrm{DualGap}(\theta_G, \theta_Q, A, \theta_f) = \max_{\overline{\theta}_f} R(\theta_G, \theta_Q, A, \overline{\theta}_f) - \min_{\overline{\theta}_G, \overline{\theta}_Q, \overline{A}} R(\overline{\theta}_G, \overline{\theta}_Q, \overline{A}, \theta_f), \tag{5.10}$$

where $\hat{R}(\theta_G, \theta_Q, A, \theta_f) = \frac{1}{m}\sum_{\mathrm{i}=1}^{m} \ell(\mathbf{x}_\mathrm{i}, \mathbf{z}_\mathrm{i}; \theta_G, \theta_Q, A, \theta_f)$ and $R(\theta_G, \theta_Q, A, \theta_f) = \mathbb{E}\hat{R}(\theta_G, \theta_Q, A, \theta_f)$. When the duality gap goes to 0, our optimization converge.

The complete algorithm is given in Algorithm 1, where back propagation is used with stochastic gradient descent techniques to optimize the objective. To be concrete, we adopt a stochastic gradient descent algorithm called the ADAM [65] to estimate the unknown parameters in neural networks. The ADAM is an algorithm for first-order gradient-based optimization of stochastic objection functions, based on adaptive estimates of lower-order moments. Given the current tuple $(\theta_G^k, \theta_Q^k, \theta_f^k, A^k)$ at the $k$-th iteration, we sample a batch of observations $\{\mathbf{x}_\mathrm{i}^k\}_{\mathrm{i}=1}^m \sim P_X$, latent variable $\{\mathbf{z}_\mathrm{i}^k\}_{\mathrm{i}=1}^m \sim P_Z$, and $\{\epsilon_\mathrm{i}\}_{\mathrm{i}=1}^m \sim U[0,1]$. Then

we construct $\hat{\mathbf{x}}_{\mathrm{i}}^k \leftarrow \epsilon_{\mathrm{i}}\mathbf{x}_{\mathrm{i}}^k + (1-\epsilon_{\mathrm{i}})G(A^k\mathbf{z}_{0,\mathrm{i}}^k;\theta_G^k)$, $\mathrm{i} = 1,...,m$ for computing the gradient penalty. Let $\hat{R}^k = \frac{1}{m}\sum_{\mathrm{i}=1}^m \ell(\mathbf{x}_{\mathrm{i}}^k, \mathbf{z}_{\mathrm{i}}^k; \theta_G^k, \theta_Q^k, A^k, \theta_f^k)$ and $J_1(\theta_f^k) = (\|\nabla_{\hat{\mathbf{x}}^k} f(\hat{\mathbf{x}}^k;\theta^k)\|_2 - 1)^2$. We can evaluate the gradient with respect to $\theta_f$, which is denoted by

$$-\nabla_{\theta_f}(\hat{R}^k + J_1(\theta_f^k)) = \nabla_{\theta_f}\left[\frac{1}{m}\sum_{\mathrm{i}=1}^m \left(f(G(A^k\mathbf{z}_{0,\mathrm{i}}^k;\theta_G^k);\theta_f^k) - f(G(Q(\mathbf{x}_{\mathrm{i}}^k;\theta_Q^k);\theta_G^k);\theta_f^k) + \lambda_1 J_1(\theta_f^k)\right)\right].$$

Then we can update $\theta_f^k$ by the ADAM using this gradient. Similarly, we can evaluate the gradient with respect to $\theta_G$, $\theta_Q$ and $A$, which is denoted by

$$\begin{aligned}
&\nabla_{\theta_G,\theta_Q,A}(\hat{R}^k + J_2(\theta_Q^k, A^k) + J_3(A^k)) \\
=&\nabla_{\theta_G,\theta_Q,\theta_A}\left[\frac{1}{m}\sum_{\mathrm{i}=1}^m \left(\|\mathbf{x}_{\mathrm{i}}^k - G(Q(\mathbf{x}_{\mathrm{i}}^k;\theta_Q^k);\theta_G^k)\| + f(G(Q(\mathbf{x}_{\mathrm{i}}^k;\theta_Q^k);\theta_G^k);\theta_f^{k+1}) - f(G(A^k\mathbf{z}_{0,\mathrm{i}}^k;\theta_G^k);\theta_f^{k+1})\right.\right. \\
&\left.\left. + \lambda_2 J_2(\theta_Q^k, A^k) + \lambda_3 J_3(A^k)\right)\right].
\end{aligned}$$

Then we can update $(\theta_G^k, \theta_Q^k, A^k)$ by the ADAM using this gradient. The stopping criteria are both the DualGap$(\theta_G^k, \theta_Q^k, A^k, \theta_f^k)$ in Equation 5.10 and the objective function $\hat{R}(\theta_G^k, \theta_Q^k, A^k, \theta_f^k)$ are less than pre-specified error tolerances $\epsilon_1$ and $\epsilon_2$, respectively. Specifically, based on the definition of the duality gap in Equation 5.10, we approximate DualGap$(\theta_G^k, \theta_Q^k, A^k, \theta_f^k)$ by the difference between $\hat{R}(\theta_G^k, \theta_Q^k, A^k, \theta_f^{k+1})$ and $\hat{R}(\theta_G^{k+1}, \theta_Q^{k+1}, A^{k+1}, \theta_f^{k+1})$.

### 5.3 Theoretical Results

From the population level, the LWGAN minimizes the new $\overline{W}_1$ divergence

$$\inf_{G\in\mathcal{G}, A\in\mathcal{A}} \overline{W}_1(P_X, P_{G(AZ_0)}), \tag{5.11}$$

which is equivalently to the optimization of the minimax problem

$$\inf_{\theta_G,\theta_Q,A}\sup_{\theta_f} \mathbb{E}\ell(X, Z; \theta_G, \theta_Q, A, \theta_f). \tag{5.12}$$

During the training, we instead minimize the empirical vision of this divergence $\overline{W}_1(\hat{P}_X, \hat{P}_{G(AZ_0)})$ based on $m$ samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ from the distribution $P_X$, and $m$ samples $\{\mathbf{z}_{0,1}, \ldots, \mathbf{z}_{0,m}\}$ from the distribution $N(0, I_d)$:

$$\inf_{G\in\mathcal{G}, A\in\mathcal{A}} \overline{W}_1(\hat{P}_X, \hat{P}_{G(AZ_0)}), \tag{5.13}$$

where

$$\overline{W}_1(\hat{P}_X, \hat{P}_{G(Z^*)}) = \inf_{Q\in\mathcal{Q}} \sup_{f\in\mathcal{F}} \Big\{ \frac{1}{m}\sum_{\mathrm{i}=1}^{m} \Big[ \|\mathbf{x}_\mathrm{i} - G(Q(\mathbf{x}_\mathrm{i}))\| + f(G(Q(\mathbf{x}_\mathrm{i}))) \Big] - \frac{1}{m}\sum_{\mathrm{i}=1}^{m} \Big[ f(G(A\mathbf{z}_{0,\mathrm{i}})) \Big] \Big\}.$$

Similarly, it equals to optimizing the empirical minimax problem

$$\inf_{\theta_G, \theta_Q, A} \sup_{\theta_f} \frac{1}{m}\sum_{\mathrm{i}=1}^{m} \ell(\mathbf{x}_\mathrm{i}, \mathbf{z}_\mathrm{i}; \theta_G, \theta_Q, A, \theta_f). \tag{5.14}$$

We are interested in answering the following two theoretical questions: Whether solving the empirical optimization problem is able to identify the correct intrinsic dimension? Whether the generator $G$ generalizes well, which means do we actually push $P_{G(AZ_0)}$ to be close to $P_X$ from the population perspective? Our theoretical analysis establishes the rank consistency, and provide an upper bound to the population divergence between $P_X$ and $P_{G(AZ_0)}$.

### 5.3.1 Estimation Consistency

Our approach seeks to identify the intrinsic dimension of $P_X$ via learning an informative prior degenerate normal distribution whose covariance matrix is a rank-$r$ matrix. Hence the optimal solutions to our problem are not singletons but set-valued as long as $\mathrm{rank}(A^*) = r$ is satisfied. We consider the estimation consistency through a distance between sets called Hausdorff distance [66]. For any two non-empty bounded subsets $S_1$ and $S_2$ of some Euclidean space, the Hausdorff distance between $S_1$ and $S_2$ is defined as

$$d_H(S_1, S_2) = \max\{\sup_{a\in S_1} d(a, S_2), \sup_{b\in S_2} d(b, S_1)\},$$

where $d(a, S_2) = \inf_{b \in S_2} \|a - b\|$ is the shortest distance from a point $a$ to the set $S_2$. The Hausdorff distance $d_H$ is a metric for the non-empty compact sets, and $d_H(S_1, S_2) = 0$ if and only if $S_1 = S_2$.

Denote $\Theta = \{(\theta_G, \theta_Q, A, \theta_f)\}$ as the set of parameters for the LWGAN. Assume that $\ell(x, z; \theta_G, \theta_Q, A, \theta_f)$ is continuous on $\Theta$. In the following part, we interchangeably use either notation $\theta$ or notation $(\theta_G, \theta_Q, A, \theta_f)$ for the elements of $\Theta$. Let us focus on the population optimization problem first. Assume that $\{(G^*, Q^*, A^*, f^*)\}$ is a set of solutions to the minimization problem Equation 5.11, that is, they satisfy $P_{Q^*(X)} = P_{A^* Z_0}$, $P_{G^*(A^* Z_0)} = P_X$ and $\text{rank}(A^*) = r$. Their corresponding parameters compose the set $\Theta^* = \{(\theta_G^*, \theta_Q^*, A^*, \theta_f^*)\}$, which is also a set of solutions to the population optimization Equation 5.12. To conveniently and comprehensively describe the optimal solutions $\Theta^*$, we denote $L(\theta_G, \theta_Q, A, \theta_f) = \mathbb{E}\ell(X, Z; \theta_G, \theta_Q, A, \theta_f)$, and the optimal value of Equation 5.12 as $V^* = \inf_{\theta_G, \theta_Q, A} \sup_{\theta_f} L(\theta_G, \theta_Q, A, \theta_f)$. We also introduce a max-function $\phi(\theta_G, \theta_Q, A) = \sup_{\theta_f} L(\theta_G, \theta_Q, A, \theta_f)$. An optimal solution $(\theta_G^*, \theta_Q^*, A^*, \theta_f^*) \in \Theta^*$ solves Equation 5.12 when it is a solution to both the inner maximization problem and outer minimization problem. Therefore, $\Theta^*$ can be expressed as

$$\Theta^* = \{(\theta_G^*, \theta_Q^*, A^*, \theta_f^*) : L(\theta_G^*, \theta_Q^*, A^*, \theta_f^*) = \sup_{\theta_f} L(\theta_G^*, \theta_Q^*, A^*, \theta_f) = \phi(\theta_G^*, \theta_Q^*, A^*)$$
$$\text{and } \phi(\theta_G^*, \theta_Q^*, A^*) = \inf_{\theta_G, \theta_Q, A} \phi(\theta_G, \theta_Q, A) = V^*\}.$$

Similarly, consider the optimal solution $\hat{\Theta}$ to the empirical minimax problem Equation 5.14. Define the sample analogues as

$$\begin{aligned}
\hat{L}_m(\theta_G, \theta_Q, A, \theta_f) &= \frac{1}{m}\sum_{\mathrm{i}=1}^{m} \ell(\mathbf{x}_\mathrm{i}, \mathbf{z}_\mathrm{i}; \theta_G, \theta_Q, A, \theta_f) \\
\hat{V}_m &= \inf_{\theta_G, \theta_Q, A} \sup_{\theta_f} \hat{L}_m(\theta_G, \theta_Q, A, \theta_f) \\
\hat{\phi}_m(\theta_G, \theta_Q, A) &= \sup_{\theta_f} \hat{L}_m(\theta_G, \theta_Q, A, \theta_f) \\
\hat{\Theta}_m &= \Big\{(\hat{\theta}_G, \hat{\theta}_Q, \hat{A}, \hat{\theta}_f) : \hat{L}_m(\hat{\theta}_G, \hat{\theta}_Q, \hat{A}, \hat{\theta}_f) = \sup_{\theta_f} \hat{L}_m(\hat{\theta}_G, \hat{\theta}_Q, \hat{A}, \theta_f) = \hat{\phi}_m(\hat{\theta}_G, \hat{\theta}_Q, \hat{A}) \\
&\qquad \text{and } \hat{\phi}_m(\hat{\theta}_G, \hat{\theta}_Q, \hat{A}) = \inf_{\theta_G, \theta_Q, A} \hat{\phi}_m(\theta_G, \theta_Q, A) = \hat{V}_m \Big\}
\end{aligned}$$

During the training, algorithms typically search for approximated solutions rather than exact solutions to Equation 5.14. We therefore allow the slackness by a power of $\tau_m$, where $\tau_m$ is a sequence of non-negative random variables such that $\tau_m \xrightarrow{p} 0$. Define

$$\begin{aligned}
\hat{\Theta}_m(\tau_m) = \Big\{(\hat{\theta}_G, \hat{\theta}_Q, \hat{A}, \hat{\theta}_f) :& \hat{L}_m(\hat{\theta}_G, \hat{\theta}_Q, \hat{A}, \hat{\theta}_f) \geq \sup_{\theta_f} \hat{L}_m(\hat{\theta}_G, \hat{\theta}_Q, \hat{A}, \theta_f) - \tau_m \\
&\text{and } \hat{\phi}_m(\hat{\theta}_G, \hat{\theta}_Q, \hat{A}) \leq \inf_{\theta_G, \theta_Q, A} \hat{\phi}_m(\hat{\theta}_G, \hat{\theta}_Q, \hat{A}) + \tau_m \Big\}
\end{aligned}$$

as the set of approximated solutions to the empirical problem Equation 5.14.

We will adopt some ideas from [67] to prove the estimation consistency. To prove the consistency under the Hausdorff distance, i.e., $d_H(\hat{\Theta}_m(\tau_m), \Theta^*) \xrightarrow{p} 0$, we need to separately show the one-sided Hausdorff consistency

$$\sup_{\theta \in \hat{\Theta}_m(\tau_m)} d(\theta, \Theta^*) \xrightarrow{p} 0 \qquad \text{and} \qquad \sup_{\theta \in \Theta^*} d(\theta, \hat{\Theta}_m(\tau_m)) \xrightarrow{p} 0.$$

The former one follows the standard proof of consistency and relies on a suitable uniform law of large numbers combined with an appropriate set-identification condition for $\Theta^*$. The latter one is based on the uniform convergence. We make some necessary assumptions on LWGAN:

**Assumptions.**

1. $\ell(\theta)$ is continuous on $\Theta$, and $\mathbb{E}[\sup_{\theta\in\Theta}|\ell(\mathbf{x},\mathbf{z};\theta)|] < \infty$.

2. LWGAN is smooth, which means the function $\ell(\mathbf{x},\mathbf{z};\theta)$ is continuously differentiable on $\Theta$ for all $(\mathbf{x},\mathbf{z}) \in \mathcal{X}\times\mathcal{Z}$ with $\mathbb{E}[\sup_{\theta\in\Theta}|\partial\ell(X,Z,\theta)/\partial\theta|^2] < \infty$.

The mild moment conditions facilitate ruling out degenerate cases, while differentiability is a common requirement for training methods employed in GAN applications.

**Theorem 5.3.1.** *Suppose* $\sup_{\theta\in\Theta} m^{1/2}|\hat{L}_m(\theta) - L(\theta)| = \mathcal{O}_p(1)$ *hold with the function* $L(\theta)$ *continuous in* $\theta$. *And suppose* $\tau_m$ *is a sequence of positive random variables such that* $\tau_m \xrightarrow{p} 0$ *and* $m^{-1/2}/\tau_m \xrightarrow{p} 0$. *Then* $d_H(\hat{\Theta}_m(\tau_m), \Theta^*) \xrightarrow{p} 0$.

Here $\mathcal{O}_p(1)$ stands for a sequence of random variables that is bounded in probability. Note that Assumption 1 on LWGAN implies that $\sup_{\theta\in\Theta}|\hat{L}_m(\theta) - L(\theta)| \xrightarrow{p} 0$ by [68], and Assumption 2 about smoothness of LWGAN indicates $|\ell(\mathbf{x},\mathbf{z},\theta_1) - \ell(\mathbf{x},\mathbf{z},\theta_2)| \leq \sup_{\theta\in\Theta}|\partial\ell(\mathbf{x},\mathbf{z},\theta)/\partial\theta||\theta_1-\theta_2|$ using mean value theorem, where $\mathbb{E}[\sup_{\theta\in\Theta}|\partial\ell(X,Z,\theta)/\partial\theta|^2] < \infty$. Thus the first assumption of Theorem Theorem 5.3.1 holds. Theorem Theorem 5.3.1 assures that the estimation of LWGAN is consistent regardless to the number of solutions, hence it is straightforward to claim that the rank of estimated $\hat{A}$ consists with the intrinsic dimension of $P_X$.

### 5.3.2 Generalization Error Bound

In the context of supervised learning, the generalization error is defined as the gap between the empirical risk (a.k.a. the training error) and the expected risk (a.k.a. the testing error). It is a measure of how accurately an algorithm is able to predict outcome values for previously unseen data. Similarly in the framework of the LWGAN, we can define its generalization error as follows [69].

**Definition 5.3.1.** *Given $\hat{P}_X$, an empirical version of the true distribution with $m$ samples, a generated distribution $P_{G(AZ_0)}$ generalizes under the divergence $\overline{W}_1(\cdot,\cdot)$ with generalization error $\epsilon$ if the following holds with high probability*

$$\left|\overline{W}_1(P_X, P_{G(AZ_0)}) - \overline{W}_1(\hat{P}_X, \hat{P}_{G(AZ_0)})\right| \leq \epsilon$$

*where $\hat{P}_{G(AZ_0)}$ is an empirical version of the generated distribution $P_{G(AZ_0)}$ with polynomial number of samples that are drawn after $P_{G(AZ_0)}$ is fixed.*

Our target is to make the former population distance small, whereas the latter empirical one is what we can access and minimize in practice. Therefore, a smaller generalization error is expected because it implies the population distance $\overline{W}_1(\cdot,\cdot)$ between the true and generated distribution is close to the empirical distance between the empirical distributions.

**Theorem 5.3.2.** *Given a fixed $L_G$-Lipschitz generator $G$, a set of 1-Lipschitz discriminator $\mathcal{F}$, and a set of decoders $\mathcal{Q}$ whose functions are $L_Q$-Lipschitz with respect to the input, and $L_{\theta_Q}$-Lipschitz with respect to its parameter $\theta_Q \in \Theta_Q$, let $\hat{\Theta}_Q$ be a $\epsilon/(16L_G L_{\theta_Q})$-net of the parameter space $\Theta_Q$ of $\mathcal{Q}$, and $\hat{\mathcal{A}}$ be a $\epsilon/(8\sqrt{5d}L_G)$-net of the parameter space $\mathcal{A}$ of $A$. Then the following inequality holds with a probability of at least*

$$1 - \mathrm{e}^{-d} - 2|\hat{\Theta}_Q||\hat{\mathcal{A}}| \exp\left\{ - \frac{\epsilon^2 m}{8(1 + 2L_G L_Q + \sqrt{5d}L_G \|A\|)^2} \right\}$$

*over the choice of $m$ samples $S_x$ from the data distribution $P_X$ and $m$ samples $S_{z_0}$ from the standard normal distribution $N(0, I_d)$,*

$$\sup_{A \in \mathcal{A}} \left|\overline{W}_1(P_X, P_{G(AZ_0)}) - \overline{W}_1(\hat{P}_X, \hat{P}_{G(AZ_0)})\right| \leq 2\mathfrak{R}_m(\mathcal{F} \circ G \circ \mathcal{Q}) + 2\mathfrak{R}_m(\mathcal{F} \circ G \circ \mathcal{A}) + \epsilon, \tag{5.15}$$

*where $\mathfrak{R}_m(\mathcal{F} \circ G \circ \mathcal{Q}) = \mathbb{E}_\delta \left\{ \sup_{f \in \mathcal{F}} m^{-1} \sum_{\mathrm{i}=1}^m \delta_\mathrm{i} f(G(Q(\mathbf{x}_\mathrm{i}))) \right\}$ is the Rademacher complexity of the function set $\mathcal{F} \circ G \circ \mathcal{Q}$, and $\mathfrak{R}_m(\mathcal{F} \circ G \circ \mathcal{A}) = \mathbb{E}_\delta \left\{ \sup_{f \in \mathcal{F}} m^{-1} \sum_{\mathrm{i}=1}^m \delta_\mathrm{i} f(G(A\mathbf{z}_{0,\mathrm{i}})) \right\}$ is the Rademacher complexity of the function set $\mathcal{F} \circ G \circ \mathcal{A}$, in which $\delta_\mathrm{i}$ is the Rademacher variable.*

Theorem Theorem 5.3.2 describes how the function classes $\mathcal{F}$, $\mathcal{Q}$ and $\mathcal{A}$ contribute to the generalization error bound in our framework. Given a fixed generator $G$, there exists a uniform upper bound for any discriminator $f \in \mathcal{F}$, decoder $Q \in \mathcal{Q}$, and low-rank matrix $A$ with appropriate numbers of samples from $P_X$ and $P_{Z_0}$. More concretely, if $|\hat{\Theta}_Q|$ and $|\hat{\mathcal{A}}|$ are small, and we have a large amount of samples, the generalization error is consequentially guaranteed to hold with a higher probability. In [70], it has been proved that $\log(|\hat{\Theta}_Q|) \leq \mathcal{O}(K_Q^2 D_Q \log(D_Q L_Q L_G L_{\theta_Q}/\epsilon))$, where $K_Q$ and $D_Q$ denote the width and the depth of $Q$ respectively, and $\log(|\hat{\mathcal{A}}|) \leq \mathcal{O}(\text{rank}(A) d \log(\sigma_{\max}\sqrt{d} L_G/\epsilon))$, where $\sigma_{\max}$ is the maximum singular value of $A$. Additionally, Lipschitz constants of networks $Q$, $G$ and $f$ are under the control of the spectral normalization of their weights. As a result, when the sample size

$$m \geq \frac{C}{\epsilon^2}\left(1 + 2L_G L_Q + \sqrt{5d} L_G \|A\|\right)^2 \left(K_Q^2 D_Q \log(D_Q L_Q L_G L_{\theta_Q}/\epsilon) + rd \log(\sigma_{\max}\sqrt{d} L_G/\epsilon)\right) \tag{5.16}$$

for some constant $C$, Equation 5.15 holds with a probability at least $1 - \exp(-K_Q^2 D_Q - rd) - \exp(d)$.

The Rademacher complexities in Equation 5.15 measure richness of a class of real-valued functions with respect to a probability distribution. There are several existing results on the Rademacher complexity of neural networks. For example, under some mild conditions, $\mathfrak{R}_m(\mathcal{F} \circ G \circ \mathcal{Q})$ is upper bounded by an order scaling as $\mathcal{O}(L_G L_Q(\sqrt{(K_Q^2 D_Q + K_f^2 D_f)/m}))$, where $K_f$ and $D_f$ denote the width and depth of the discriminator $f$, and an upper bound on $\mathfrak{R}_m(\mathcal{F} \circ G \circ \mathcal{A})$ scales as $\mathcal{O}(L_G \sigma_{\max}(\sqrt{(rd + K_f^2 D_f)/m}))$ [70]. Combining this information with Equation 5.16 and plugging into Equation 5.15, we are able to conclude that for some constant $C_{model}$ based on the model property

$$\sup_{A \in \mathcal{A}} \left| \overline{W}_1(P_X, P_{G(AZ_0)}) - \overline{W}_1(\hat{P}_X, \hat{P}_{G(AZ_0)}) \right| \leq C_{model}\epsilon$$

with probability at least $1 - \exp(-d) - \exp(-K_Q^2 D_Q - rd)$.

Since $\overline{W}_1(P_X, P_{G(AZ_0)})$ is a tight upper bound on the 1-Wasserstein distance between $P_X$ and $P_{G(AZ_0)}$ from Theorem 5.2.1, we further have

$$W_1(P_X, P_{G(AZ_0)}) \leq \overline{W}_1(\hat{P}_X, \hat{P}_{G(AZ_0)}) + 2\mathfrak{R}_m(\mathcal{F} \circ G \circ \mathcal{Q}) + 2\mathfrak{R}_m(\mathcal{F} \circ G \circ \mathcal{A}) + \epsilon$$

with high probability. This implies that from the population perspective, the real distribution is close to the generated distribution with respective to 1-Wasserstein distance when we minimize our loss function $\overline{W}_1(\hat{P}_X, \hat{P}_{G(AZ_0)})$ in practice.

## 5.4 Experimental Results

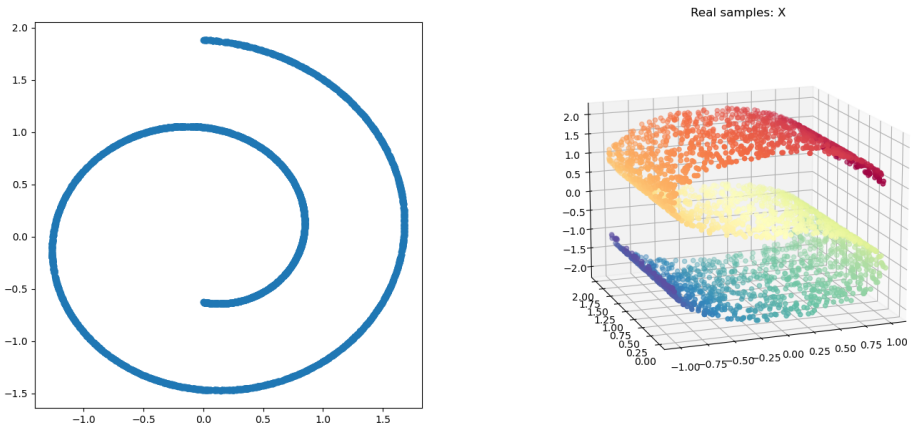
In this section, we will demonstrate a comprehensive numerical experiments which validate that the LWGAN is able to reach the our four goals simultaneously: detecting the correct intrinsic dimension, generating high-quality samples, decoding meaningful latent codes, and obtaining small reconstruction errors. Our codes in PyTorch are available at https://drive.google.com/drive/folders/1piLXjguswG0Nn_npAFgAW-74uzkqRUjs?usp=sharing.

### 5.4.1 Toy Data

We first verify our method using three toy examples supported on smooth manifold with increasing dimensions. Besides the S-curve data we mentioned in the Section 2, the other two datasets are described as:

1. Swiss Roll: $x_1 = \frac{3\pi(1+2\mathrm{i})}{2}\cos(\frac{3\pi(1+2\mathrm{i})}{2})$, $x_2 = \frac{3\pi(1+2\mathrm{i})}{2}\sin(\frac{3\pi(1+2\mathrm{i})}{2}))$, for $\mathrm{i} \sim N(0,1)$.
2. Hyperplane: $x_1, x_2, x_3, x_4 \sim N(0,1)$, $x_5 = x_1 + x_2 + x_3 + x_4^2$.

It is straightforward that the intrinsic dimension of the Swiss Roll, S-curve and Hyperplane are 1, 2, 4 respectively. In particular, if we embed the images of S-curve in $\mathbb{R}^4$ and that of Hyperplane in $\mathbb{R}^8$, they can be represented as graphs of continuous functions, so that an identity mapping $G \circ Q$ exists according to Theorem Theorem 5.1.1 for these two cases. For the training, all models are trained with a batch size 256 for 16k iterations. Two different dimension of the latent space $d = \{5, 10\}$ are adapted to confirm that our approach




(a) Swiss Roll (b) S-curve

**Figure 5.3.** Two Manifolds

is stable to pick the right intrinsic dimension of the data manifold $\mathcal{X}$. We further train 4 LWGANs on each dimension of $Z$ with $\lambda_1 = 5.0$, $\lambda_2 = 10.0$, and a variety of values for $\lambda_3 = \{0, 0.001, 0.01, 0.1\}$. All other model structures and training parameters are the same for these three datasets.

**Intrinsic Dimension** For each dataset, we compare the performance on different choices of $\lambda_3$s with respect to the test reconstruction errors, MMDs between $Q(X)$ and $Z$ as well as MMDs between $G(Z)$ and $X$ obtained at the end of training. The optimal $\lambda_3$ for each dimension of the latent space is selected to have the smallest reconstruction error and MMDs. Therefore, the best $\lambda_3$s for Swiss Roll with two latent dimensions $d = 5$ and $d = 10$ are the same to be 0.1; in terms of S-curve, the optimal $\lambda_3$s are also identical for $d = 5$ and $d = 10$, and chosen as 0.01; for Hyperplane, we select $\lambda_3 = 0.001$ when $d = 5$, and $\lambda_3 = 0.01$ when $d = 10$. The eigenvalues of $AA^T$ at the best $\lambda_3$ from the second column in Figure 5.4 show that our approach enables to detect the correct intrinsic dimension of the data manifold no matter what the dimension of the latent space is. This implies that our model can tell that the intrinsic dimensions of these three datasets are is 1, 2 and 4 respectively. Meanwhile, the last two columns in Figure 5.4 conveys that we are able to generate high quality sample using the latent variable from $N(0, AA^T)$ and recover the original data $X$ using the encodes $Q(X)$.

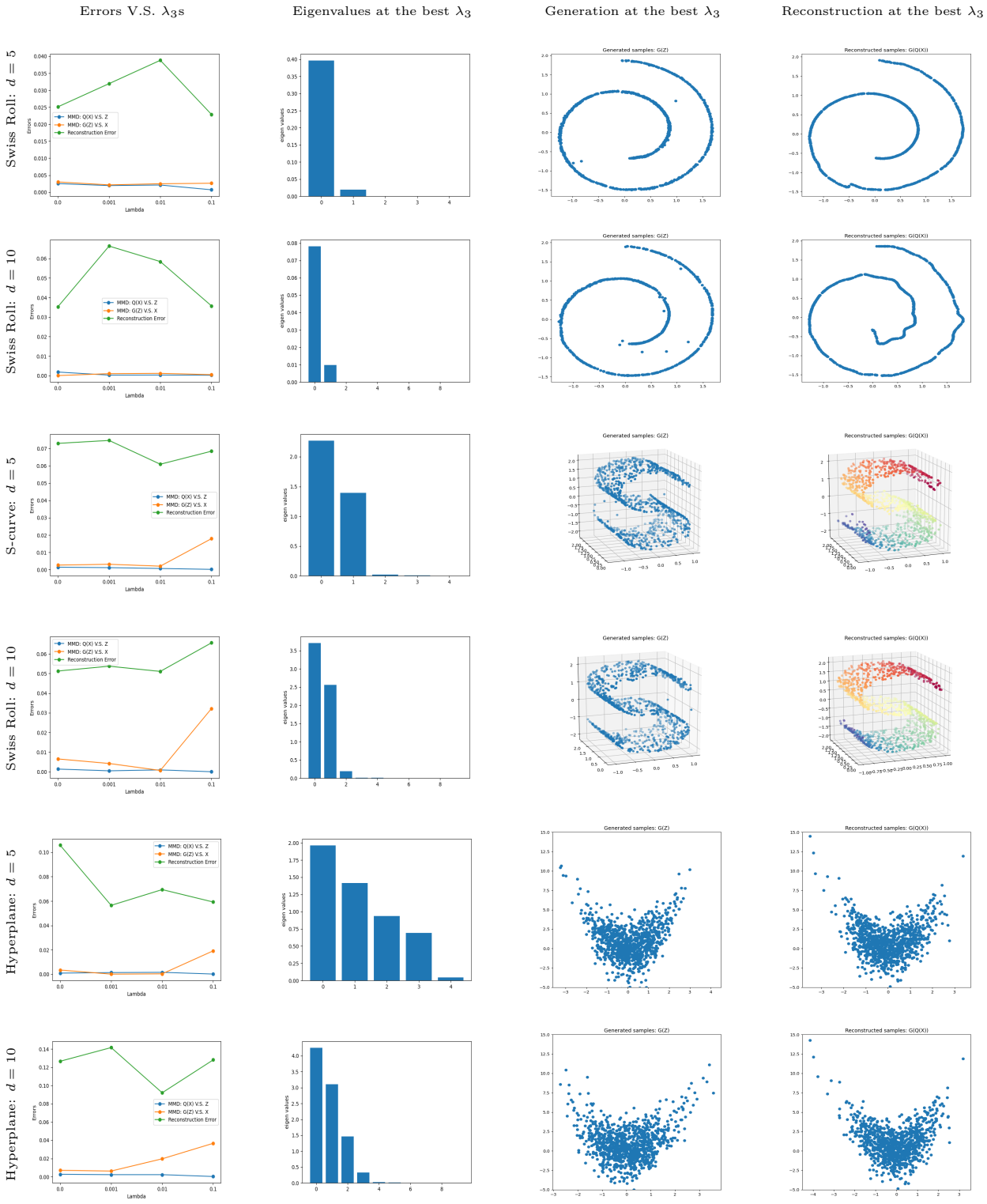
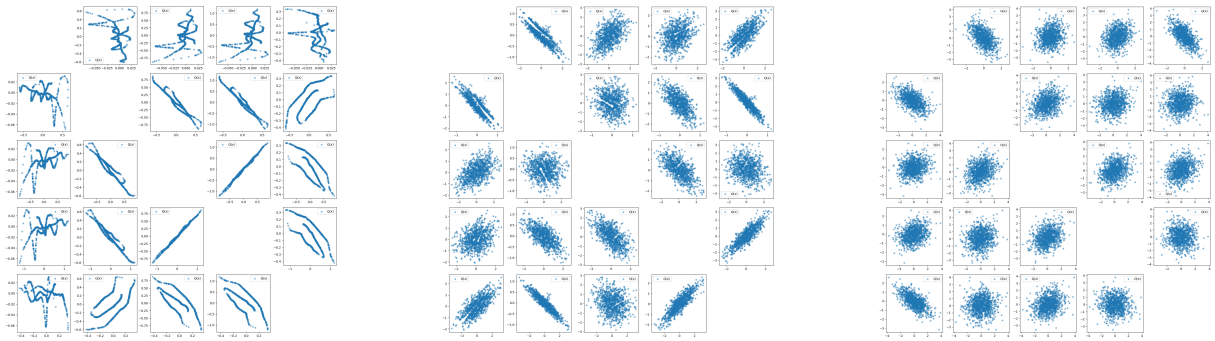



**Figure 5.4.** Toy Datasets: The first column plots the relationship between the regularisation power $\lambda_3$s and the errors of each model. The second column shows the eigenvalues at the optimal $\lambda_3$. The third column is the generated data. And the last column shows the testing reconstructions.

**Latent space** After training, the distribution of $Q(X)$ is expected to be close to the distribution of $Z$, which is confirmed by MMDs between $Q(X)$ and $Z$ in the first column of Figure 5.4. We visually demonstrate the 5-dimensional latent distribution of $Q(X)$ at the best $\lambda_3$ for above three toy datasets. Specifically, we plot the ith component of $Q(X)$, $Q(X)_\mathrm{i}$, against the jth component of $Q(X)$, $Q(X)_\mathrm{j}$, for all $\mathrm{i} \neq \mathrm{j}$ in Figure 5.5. We can tell that there exist correlations of different strength between any two dimensions of $Q(X)$. Some of them are highly correlated with a coefficient close to 1, while some of them are almost uncorrelated. It validates that the encodes $Q(X)$ follows a degenerated normal distribution $N(0, A^*A^{*T})$. We also noticed that although the latent space of the Swiss Roll shows the shape of a normal distribution, their points are not evenly scattered as the other two. The possible explanation is that the images of Swiss Roll in $\mathbb{R}^2$ cannot be expressed as a graph of a continuous function, so its corresponding $Q^*$ is a surjective function mapping multiple inputs to one output, which results in clustering points in the latent space.



(a) Swiss Roll (b) S-curve (c) Hyperplane

**Figure 5.5.** Toy Datasets: Latent space in $\mathbb{R}^5$

**Datasets on non-smooth manifold** Although our approach assumes that the data lies on a smooth manifold, we would like to check if our model works for noncontinuous datasets that are parametrised by a finite number of parameters. Then the intrinsic dimension refers to the minimal number of parameters needed. We consider two datasets described by mixture of Gaussians: (a). RING: a mixture of 8 Gaussians with means $\{(2 \cdot \cos\frac{2\pi\mathrm{i}}{8}, 2 \cdot \sin\frac{2\pi\mathrm{i}}{8}) | \mathrm{i} = 0, \dots 7\}$ and standard deviation 0.02; (b). GRID: a mixture of 25 Gaussians with means $\{(2 \cdot \mathrm{i}, 2 \cdot \mathrm{j}) | \mathrm{i} = -2, -1, \dots, 2, \mathrm{j} = -2, -1, \dots, 2\}$ and standard deviation 0.02. We observe

that our model can successfully identify the intrinsic dimension of the RING is 1 and that of GRID is 2 from their eigenvalues in Figure 5.6. At the same time, the generator $G$ is able to produce high-quality data using latent variables from $N(0, AA^T)$. The reconstructions between $X$ and $G(Q(X))$ also perform well.

### 5.4.2 MNIST

The MNIST database (Modified National Institute of Standards and Technology database) is a large database of handwritten digits $0 \sim 9$ that is commonly used for training various image processing systems. The MNIST database contains $70,000$ $28 \times 28$ grey images. It was shown that different digits have different intrinsic dimension [71]. Hence the distribution of MNIST may be supported on several disconnected manifold with various intrinsic dimensions.

We first train models on digit 1 and digit 2 separately using a 64-dimensional latent variable. The first column of Figure 5.7 plots the eigenvalues of learned $AA^T$ for these two digits. It can be observed the eigenvalues smaller than the 4th largest one diminish to 0 for digit 1, while those eigenvalues converge to 0 after the 13th largest eigenvalue for digit 2. Therefore, our estimation of the intrinsic dimension for digit 1 is smaller than the one in [71] whose estimation is 8, but these two estimations for digit 2 match to be around 13. The generative digits and reconstructed digits are presented in the last two columns of Figure 5.7.

Previously we have confirmed that our model can be used to datasets that are not be supported on a smooth manifold using mixture of Gaussians. We further check it with all digits from MNIST. The intrinsic dimension of MNIST is suggested to be about 13 from Figure 5.8a. Corresponding generated samples and reconstructed samples are provided in Figure 5.8b and Figure 5.8c. Interpolation on the latent space between two digits given in Figure 5.8d establishes that our model can get rid of mode collapsing. In particular, we sample pairs of testing examples $x_1$ and $x_2$ and project them into $z_1$ and $z_2$ by the encoder $Q$. We then linearly interpolate between $z_1$ and $z_2$ and pass the intermediary points through the generator $G$ to plot the input-space interpolations.

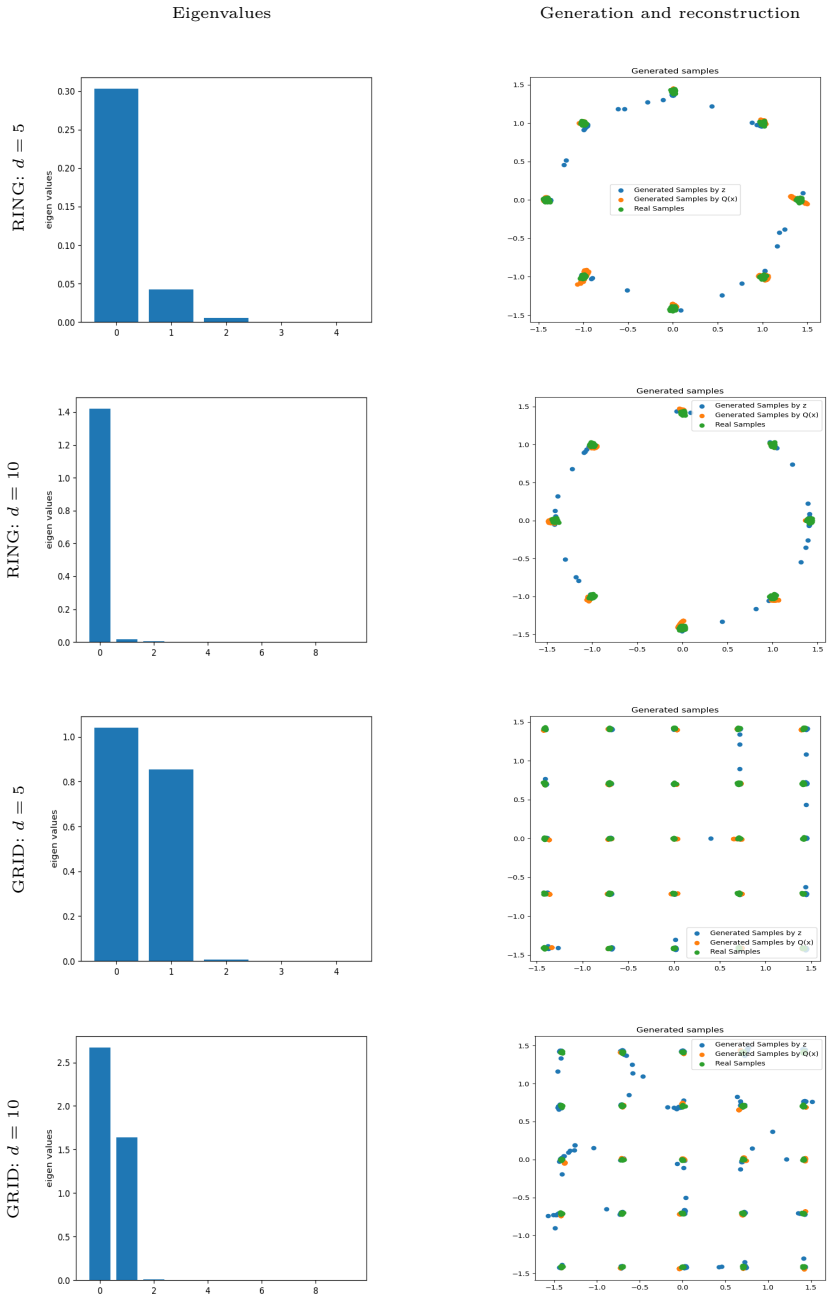



**Figure 5.6.** Mixture of Gaussians: The first column show the eigenvalues of $AA^T$ with different dimensions. The second column show the reconstructed samples $G(Q(X))$ and generated samples $G(Z)$, where $Z \sim N(0, AA^T)$.

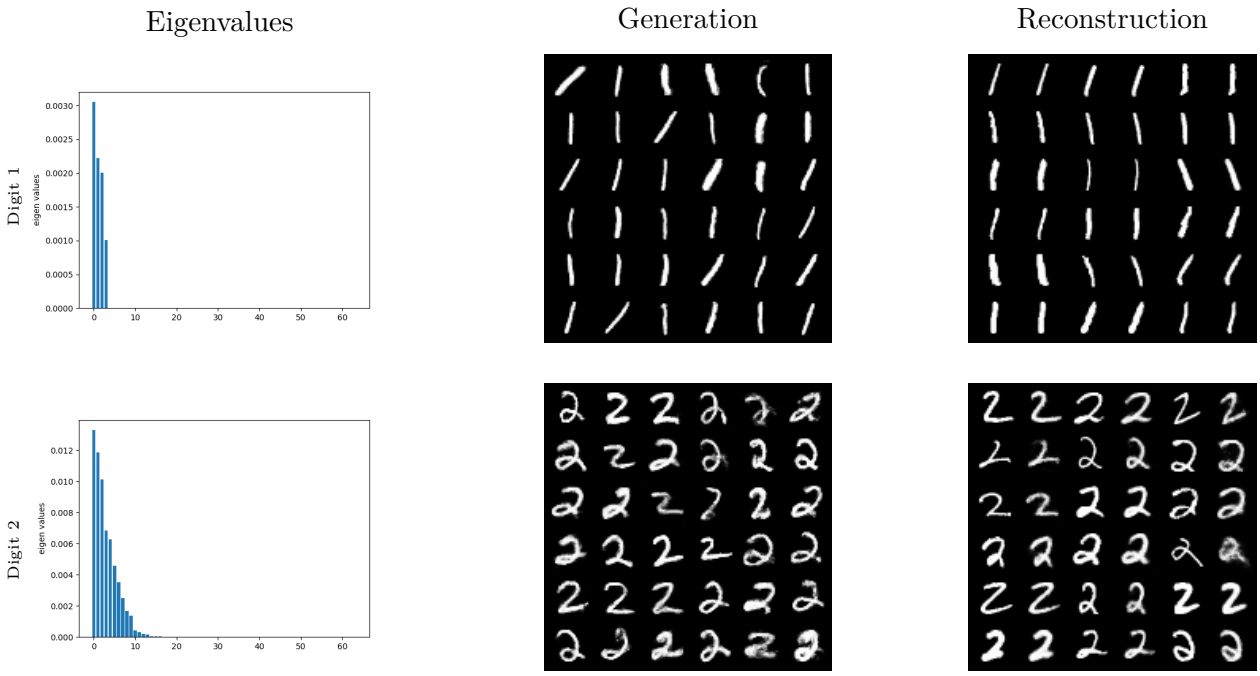
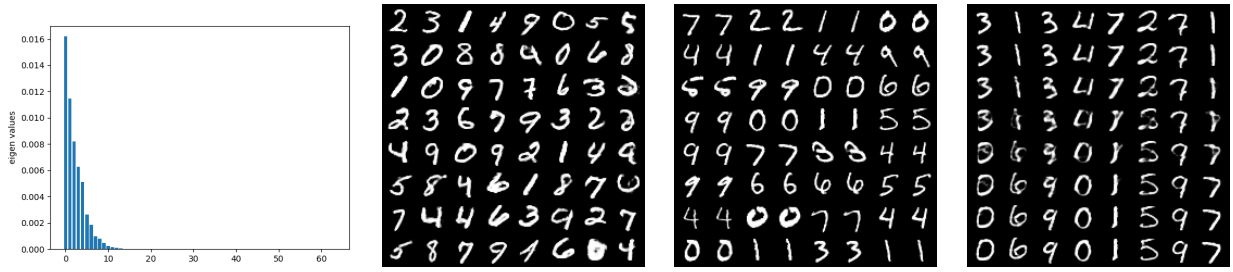



**Figure 5.7.** Digits 1 and 2: The first column are the eigenvalues of digit 1 and digit 2, the second column presents the generating samples of digit 1 and digit 2, the third column are reconstructed samples of digit 1 and digit 2.




(a) Eigenvalues (b) Generation (c) Reconstruction (d) Interpolation

**Figure 5.8.** MNIST: Results of LWGAN on 64-dimensional latent space.

### 5.4.3 CelebA

CelebA (CelebFaces Attributes Dataset) is another ideal benchmark datasets for training models to generate synthetic images. It is a large-scale face attributes dataset with $202,599$ $64 \times 64$ colored celebrity face images, which cover large pose variations and diverse people.

CelebA is a more complex dataset than MNIST, and is available at http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html.

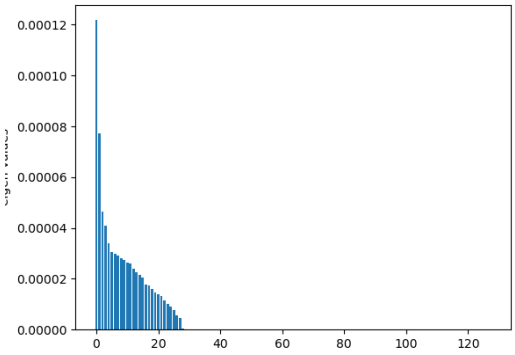
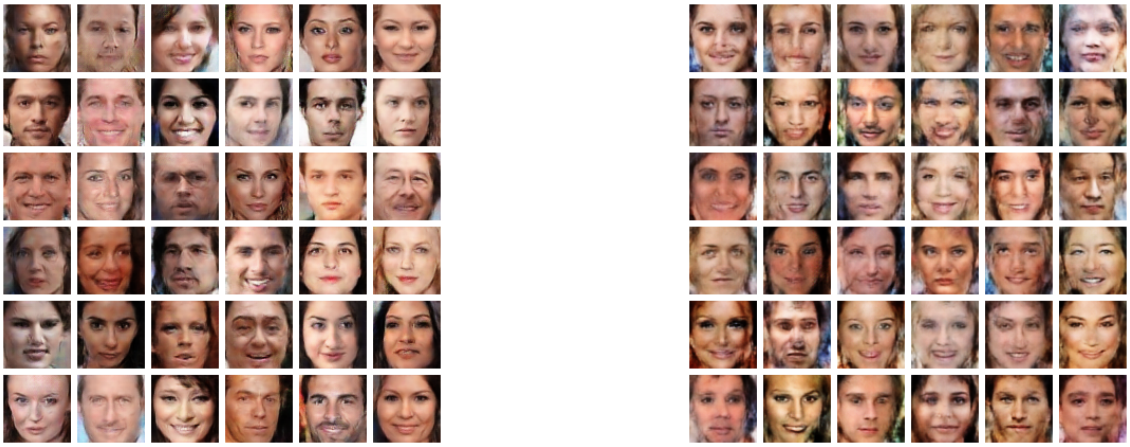



**Figure 5.9.** CelebA: Eigenvalues of $AA^T$

We train CelebA using a latent dimension $d = 128$. Figure 5.9 demonstrates that the eigenvalues of CelebA are close to 0 after the 20th largest eigenvalue, so the intrinsic dimension of CelebA is at least 20. We further compare our method with WGAN, WAE and iWGAN visually and numerically. Here iWGAN refers to our preprint work where $A$ is fixed as the identity matrix during the training. The generative faces from four methods are demonstrated in Figure 5.10. Our LWGAN is able to generate images with higher qualities than other four methods, and we notice that the images generated by WAE are very blurry due to the dimensional mismatch between $P_{Q(X)}$ and $P_Z$. Figure 5.11b provides the comparisons between reconstructed faces through $G(Q(X))$ and real faces. Note that WGAN cannot provide reconstructed images since it does not produce the latent codes. Figure 5.11a shows the interpolation between two faces of LWGAN, iWGAN and WAE.

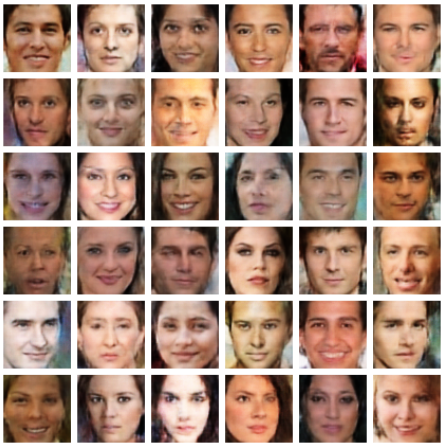
We numerically compare these methods with respect to four metrics, including inception scores (IS), Frechet inception distances (FID), reconstruction errors (RE), and maximum mean discrepancy (MMD) between encodes and normal random variables.



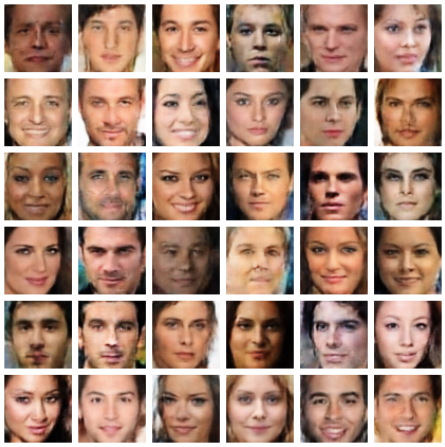
(a) WGAN



(b) WAE



(c) iWGAN



(d) LWGAN

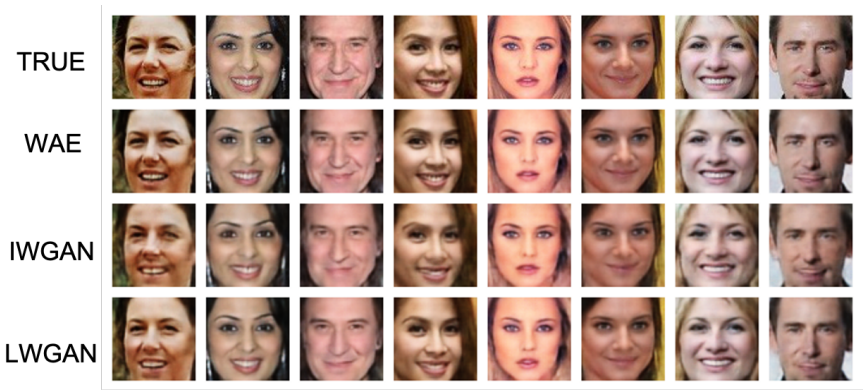
**Figure 5.10.** CelebA: Generation by different methods

**IS.** Proposed by [72], the IS involves using a pre-trained Inception v3 model to predict the class probabilities for each generated image. These predictions are then summarized into the IS by the KL divergence as following,

$$\text{IS} = \exp\left(\mathbb{E}_{\mathbf{x}\sim P_{G(Z)}} D_{KL}\left(p(y|\mathbf{x})\|p(y)\right)\right), \tag{5.17}$$

where $p(y|x)$ is the predicted probabilities conditioning on the generated images, and $p(y)$ is the corresponding marginal distribution. Higher scores are better, corresponding to a larger KL-divergence between the two distributions.


(a) Interpolation




(b) Reconstruction

**Figure 5.11.** CelebA: Interpolation and reconstruction by different methods

**FID.** The FID is proposed by [73] to improve the IS by actually comparing the statistics of generated samples to real samples. It is defined as the Fréchet distance between two multivariate Gaussians,

$$\text{FID} = \|\mu_r - \mu_G\|^2 + \text{Tr}\left(\Sigma_r + \Sigma_G - 2(\Sigma_r\Sigma_G)^{1/2}\right), \tag{5.18}$$

where $X_r \sim N(\mu_r, \Sigma_r)$ and $X_G \sim N(\mu_G, \Sigma_G)$ are the 2048-dimensional activations of the Inception-v3 pool-3 layer for real and generated samples respectively. For the FID, the lower the better.

**Reconstruction Error.** The reconstruction error (RE) is defined as

$$\text{RE} = \frac{1}{N}\sum_{\text{i}=1}^{N} \|\hat{\mathbf{x}}_\text{i} - \mathbf{x}_\text{i}\|_2, \tag{5.19}$$

**Table 5.1.** Comparison of LWGAN, iWGAN, WAE, WGAN-GP

| Methods | IS | FID | RE | MMD |
|---|---|---|---|---|
| True | 1.91(0.19) | 25.52 | – | – |
| LWGAN | **1.61(0.10)** | **43.99** | **10.31(1.94)** | $\mathbf{1.8 \times 10^{-3}}$ |
| iWGAN | 1.56(0.06) | 45.60 | 12.06(3.25) | $4.1 \times 10^{-3}$ |
| WAE | 1.43(0.09) | 54.35 | **7.09(0.99)** | $2.3 \times 10^{-3}$ |
| WGAN | **1.64(0.11)** | **39.93** | – | – |

where $\hat{X}_{\mathrm{i}}$ is the reconstructed sample for $X_{\mathrm{i}}$. RE is used to measure if the method has generated meaningful latent encodings. Smaller reconstruction errors indicate a more meaningful latent space which can be decoded into the original samples.

**MMD.** MMD is defined as

$$\mathrm{MMD} = \frac{1}{N(N-1)} \sum_{l \neq \mathrm{j}} \kappa(\mathbf{z}_l, \mathbf{z}_{\mathrm{j}}) + \frac{1}{N(N-1)} \sum_{l \neq \mathrm{j}} \kappa(\tilde{\mathbf{z}}_l, \tilde{\mathbf{z}}_{\mathrm{j}}) - \frac{2}{N^2} \sum_{l,\mathrm{j}} \kappa(\mathbf{z}_l, \tilde{\mathbf{z}}_{\mathrm{j}}) \tag{5.20}$$

where $\kappa$ is a positive-definite reproducing kernel, $\mathbf{z}_{\mathrm{i}}$'s are drawn from prior distribution $P_{AZ_0}$, and $\tilde{\mathbf{z}}_{\mathrm{i}} = Q(\mathbf{x}_{\mathrm{i}})$ are the latent encodings of real samples. MMD is used to measure the difference between distribution of latent encodings and standard normal random variables. Smaller MMD indicates that the distribution of encodings is close to the standard normal distribution.

Table 5.1 shows that in terms of the performance of generative models, LWGAN is slightly worse than the WGAN due to its sacrifice to the detection of intrinsic dimension, but is significantly better than WAE that suffers from generating clear faces. In terms of reconstruction and similarities on the latent space, LWGAN can achieve comparable results to WAE. Overall, our LWGAN enable to successfully detect the correct intrinsic dimension, produce meaningful encodes and reliable images at the same time.

## 5.5 Related Proofs

### 5.5.1 Proof of Theorem 5.1.1

According to the Whitney embedding theorem, for every $d$-dimensional compact smooth manifold $\mathcal{X}$, there exists an embedding $u : \mathcal{X} \to \mathbb{R}^{2d}$ from $\mathcal{X}$ to $\mathbb{R}^{2d}$. Since $u$ is a diffeomorphism onto its image $u(\mathcal{X})$, $u(\mathcal{X})$ is also a $d$-dimensional submanifold embedded in $\mathbb{R}^{2d}$, and its inverse mapping $u^{-1} : u(\mathcal{X}) \to \mathcal{X}$ exists. Then we will construct $Q^* : \mathcal{X} \to \mathcal{Z}$ from the input domain $\mathcal{X}$ to a degenerated normal distribution by applying the well-known Rosenblatt transformation [74] to $u(X)$, which can transform an absolutely continuous $d$-variate distribution into the uniform distribution on the $d$-dimensional hypercube.

Let $\tilde{X} = u(X) \in \mathbb{R}^{2d}$, and write $\tilde{X} = (\tilde{X}_1, \ldots, \tilde{X}_{2d})$. Suppose the manifold $u(\mathcal{X})$ embedded in $\mathbb{R}^{2d}$ is globally a graph of the continuous function $h : V \to \mathbb{R}^d$, where $V \subseteq \mathbb{R}^d$ is an open subset. This indicates that $u(\mathcal{X})$ is a subset of $\mathbb{R}^p$ defined by

$$P = \{(\tilde{X}^{(1)}, \tilde{X}^{(2)}) \in \mathbb{R}^d \times \mathbb{R}^d : \tilde{X}^{(1)} \in V \text{ and } \tilde{X}^{(2)} = h(\tilde{X}^{(1)})\}.$$

Let $\boldsymbol{\pi}_1 : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ denote the projection onto the first factor, and let $\psi : P \to V$ be the restriction of $\boldsymbol{\pi}_1$ to $P$:

$$\psi(\tilde{X}^{(1)}, \tilde{X}^{(2)}) = \tilde{X}^{(1)}, \quad (\tilde{X}^{(1)}, \tilde{X}^{(2)}) \in P.$$

Then $\psi$ is a continuous map and a homeomorphism since its continuous inverse is given by $\psi^{-1}(\tilde{X}^{(1)}) = (\tilde{X}^{(1)}, h(\tilde{X}^{(1)}))$.

Next, we would like to transform $\tilde{X}^{(1)}$ to a $d$-dimensional variable from the uniform distribution. Denote the marginal cdfs as $F_{\mathrm{i}}(x) = \mathbb{P}(\tilde{X}_{\mathrm{i}} \leq x)$, for $\mathrm{i} = 1, \ldots, d$. By applying the probability integral transformation to each component, the random vector

$$\Big(U_1, U_2, \ldots, U_d\Big) := \Big(F_1(\tilde{X}_1), F_2(\tilde{X}_2), \ldots, F_d(\tilde{X}_d)\Big) \tag{5.21}$$

has uniformly distributed marginals. Let $C : [0,1]^d \to [0,1]$ be the copula of $\tilde{X}$, which is defined as the joint cdf of $(U_1, \ldots, U_d)$:

$$C(u_1, u_2, \ldots, u_d) = \mathbb{P}\Big(U_1 \leq u_1, U_2 \leq u_2, \ldots, U_d \leq u_d\Big). \tag{5.22}$$

The copula $C$ contains all information on the dependence structure among the components of $\tilde{X}$, while the marginal cumulative distribution functions $F_{\mathrm{i}}$ contain all information on the marginal distributions. Therefore, the joint cdf of $\tilde{X}$ is $C\Big(F_1(\tilde{\mathbf{x}}_1), F_2(\tilde{\mathbf{x}}_2), \ldots, F_d(\tilde{\mathbf{x}}_d)\Big)$. Denote the conditional distribution of $U_k$, given $U_1, \ldots, U_{k-1}$, by

$$C_k(u_k|u_1, \ldots, u_{k-1}) = \mathbb{P}\Big(U_k \leq u_k | U_1 = u_1, \ldots, U_{k-1} = u_{k-1}\Big) \tag{5.23}$$

for $k = 2, \ldots, d$. Then $d$ independent uniform random variables can be defined by

$$\begin{cases} \tilde{U}_1 = U_1, \\ \tilde{U}_k = C_k\Big(U_k | U_1, \ldots, U_{k-1}\Big), \quad k = 2, \ldots, d. \end{cases} \tag{5.24}$$

We can readily show that $\tilde{U}_1, \ldots, \tilde{U}_d$ are independent uniform random variables. This is because

$$\begin{aligned} \mathbb{P}(\tilde{U}_k \leq \tilde{u}_k : k = 1, \ldots, d) &= \int_{C_1\left(v_1\right) \leq \tilde{u}_1} \cdots \int_{C_p\left(v_d | v_1, \ldots, v_{d-1}\right) \leq \tilde{u}_d} dC_d\Big(v_d | v_1, \ldots, v_{d-1}\Big) \cdots dC_1\Big(v_1\Big) \\ &= \int_0^{\tilde{u}_1} \cdots \int_0^{\tilde{u}_d} d\mathbf{z}_d \cdots d\mathbf{z}_1 = \prod_{k=1}^{d} \tilde{u}_k. \end{aligned}$$

Finally, let $Z'_{\mathrm{i}} = \Phi^{-1}(\tilde{U}_{\mathrm{i}})$ for $\mathrm{i} = 1, \ldots, d$, where $\Phi^{-1}$ is the inverse cdf of a standard normal random variable. We further modify this $d$-dimensional continuous random variable $Z' \sim N(0, I_d)$ to a $p$-dimensional $Z$ through the linear transformation,

$$Z = T_1(Z') = (I_d \ B)^T Z',$$

such that $Z$ is from the degenerated multivariate normal distribution $N(0, AA^T)$, and $AA^T$ is a singular matrix with $\mathrm{rank}(AA^T) = d < p$. Here $B \in \mathbb{R}^{d \times (p-d)}$ is a matrix and

$AA^T = (I_d \ B)^T(I_d \ B)$ is a singular matrix. This completes the transformation $Q^*$ from $X$ to a variable $Z = (Z_1, \ldots, Z_p)$ from the $p$-dimensional degenerated normal distribution $N(0, AA^T)$.

We can get $G^* : \mathcal{Z} \to \mathcal{X}$ by reversing the above transformation. we first reverse $p$-dimensional $Z$ to a $d$-dimensional $Z'$ from the standard normal distribution by

$$Z' = T_2(Z) = (I_d \ 0_{d\times(p-d)})Z$$

Then, $Z'$ can be transformed to $d$ independent uniform random variables by $\tilde{U}_{\mathrm{i}} = \Phi(Z'_{\mathrm{i}})$ for $\mathrm{i} = 1, \ldots, d$. Next, define

$$\begin{cases} U_1 = \tilde{U}_1, \\ U_k = C_k^{-1}(\tilde{U}_k|\tilde{U}_1, \ldots, \tilde{U}_{k-1}), \quad \mathrm{i} = 2, \ldots, d, \end{cases}$$

where $C_k^{-1}(\cdot|u_1, \ldots, u_k)$ is the inverse of $C_k$ and can be obtained by numerical root finding. Then let $\tilde{X}_{\mathrm{i}}^{(1)} = F_{\mathrm{i}}^{-1}(U_{\mathrm{i}})$ for $\mathrm{i} = 1, \ldots, d$. Since any point from $u(\mathcal{X})$ can be expressed as $(\tilde{X}^{(1)}, h(\tilde{X}^{(1)}))$, and $\psi$ is a bijective function, we have $\tilde{X} = (\tilde{X}^{(1)}, \tilde{X}^{(2)}) = \psi^{-1}(\tilde{X}^{(1)})$, and finally $X = u^{-1}(\tilde{X})$. This completes the transformation $G^*$ from $Z$ to $X$, and hence $G^*(Q^*(X)) = X$.

### 5.5.2 Proof of Corollary 5.1.1

Note that any $d$-dimensional manifold embedded in $\mathbb{R}^{2d}$ locally looks like the graph of a continuous mapping from $\mathbb{R}^d$ to $\mathbb{R}^d$. That is, every point of $u(\mathcal{X})$ lies in an open subset $S \subseteq \mathbb{R}^{2d}$ such that $P = S \cap u(\mathcal{X})$ is a $d$-dimensional patch. Therefore, there exists a continuous function $h_P : V \to \mathbb{R}^d$ defined on the open set $V \subseteq \mathbb{R}^d$, such that

$$P = \{(\tilde{X}^{(1)}, \tilde{X}^{(2)}) \in \mathbb{R}^d \times \mathbb{R}^d : \tilde{X}^{(1)} \in V \text{ and } \tilde{X}^{(2)} = h_P(\tilde{X}^{(1)})\}.$$

Let $\boldsymbol{\pi}_1 : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ denote the projection onto the first factor, and let $\psi_P : P \to V$ be the restriction of $\boldsymbol{\pi}_1$ to $P$:

$$\psi_P(\tilde{X}^{(1)}, \tilde{X}^{(2)}) = \tilde{X}^{(1)}, \quad (\tilde{X}^{(1)}, \tilde{X}^{(2)}) \in P,$$

which is a continuous map. $(P, \psi_P)$ is called a chart for the manifold $u(\mathcal{X})$. Because every $d$-dimensional manifold admits a cover by $d+1$ charts, i.e., $u(\mathcal{X}) = \cup_{\mathrm{j}=1}^{d+1} P_{\mathrm{j}}$, we will have $d+1$ different mappings $h_{P_{\mathrm{j}}} : V \to \mathbb{R}^d$, which maps $\tilde{X}^{(1)}$ to different $\tilde{X}^{(2)}$s. Now, for any point $\tilde{X} \in u(\mathcal{X})$, we may define a function $\psi : u(\mathcal{X}) \to V$ as follows

$$\psi(\tilde{X}) = \sum_{\mathrm{j}=1}^{d+1} \mathbb{I}\{\tilde{X} \in P_{\mathrm{j}}, \tilde{X}^{(2)} = h_{P_{\mathrm{j}}}(\tilde{X}^{(1)})\} \psi_{P_{\mathrm{j}}}(\tilde{X}^{(1)}, h_{p_{\mathrm{j}}}(\tilde{X}^{(1)})) = \tilde{X}^{(1)},$$

where $\mathbb{I}$ is the indicator function, and let $\psi_{P_{\mathrm{j}}}(\tilde{X}^{(1)}, h_{P_{\mathrm{j}}}(\tilde{X}^{(1)})) = \tilde{X}^{(1)}$ for $\tilde{X} \in P_{\mathrm{j}}$. Then similarly to the proof of Theorem 1, we apply the transformation Equation 5.21, Equation 5.22, Equation 5.23 and Equation 5.24 in the above proof to $\tilde{X}^{(1)}$, we have $\tilde{U}_{\mathrm{i}}$, for $\mathrm{i} = 1, \ldots, d$ being independently from the uniform distribution. Finally a $p$-dimensional random variable from the degenerated normal distribution $N(0, AA^T)$ is obtained by let $Z'_{\mathrm{i}} = \Phi^{-1}(\tilde{U}_{\mathrm{i}})$ and $Z = T_1(Z')$. This completes the transformation $Q^*$ from $X$ to $Z = (Z_1, \ldots, Z_p)$.

On the opposite direction of transforming $Z$ to $X$, we first reverse $Z_{\mathrm{i}}$ for $\mathrm{i} = 1, \ldots, p$ to $\tilde{X}^{(1)}$ based on the reverse procedure in the proof of Theorem 1. Transforming $\tilde{X}^{(1)}$ to the original $\tilde{X}$ is not guaranteed since $\Psi(\tilde{X})$ is a surjective function, and $d+1$ functions $h_{P_{\mathrm{j}}}$ may map $\tilde{X}^{(1)}$ to a different $\tilde{X}^{(2)}$. However, we can construct a stochastic transformation from $\tilde{X}^{(1)}$ to $\hat{\tilde{X}}^{(2)}$:

$$\hat{\tilde{X}}^{(2)} = \mathrm{e}_P \odot (h_{P_1}(\tilde{X}^{(1)}), \ldots, h_{P_{d+1}}(\tilde{X}^{(1)})),$$

where $\mathrm{e}_P$ is a $d+1$-dimensional vector with one element being 1 and others being 0, $\odot$ refers to the Hadamard product. The location of element 1 can be determined by a multinormial distribution whose probabilities are based on the frequency of a set of samples from $\mathcal{X}$. In this way, $G^*(Z) = u^{-1}(\tilde{X}^{(1)}, \hat{\tilde{X}}^{(2)})$ follows the same distribution as $X$.

### 5.5.3 Proof of Theorem 5.2.1

By the LWGAN objective Equation 5.5, Equation 5.6 holds. Since $W_1$ is a distance between two probability measures, $W_1(P_X, P_{G(AZ_0)}) \leq \overline{W}_1(P_X, P_{G(AZ_0)})$. If there exists a $Q^* \in \mathcal{Q}$ such that $Q^*(X)$ has the same distribution as $P_{AZ_0}$, we have

$$\overline{W}_1(P_X, P_{G(AZ_0)}) \leq W_1(P_X, P_{G(Q^*(X))}) + W_1(P_{G(Q^*(X))}, P_{G(AZ_0)}) = W_1(P_X, P_{G(AZ_0)}).$$

Hence, $W_1(P_X, P_{G(AZ_0)}) = \overline{W}_1(P_X, P_{G(AZ_0)})$.

### 5.5.4 Proof of Theorem 5.3.1

We introduce the function $Q(\theta)$ and $\hat{Q}_m(\theta)$:

$$Q(\theta) = Q(\theta_G, \theta_Q, A, \theta_f) = \sup\{\phi(\theta_G, \theta_Q, A) - L(\theta_G, \theta_Q, A, \theta_f), \phi(\theta_G, \theta_Q, A) - V^*\}.$$
$$\hat{Q}_m(\theta) = \hat{Q}_m(\theta_G, \theta_Q, A, \theta_f) = \sup\{\hat{\phi}_m(\theta_G, \theta_Q, A) - \hat{L}_m(\theta_G, \theta_Q, A, \theta_f), \hat{\phi}_m(\theta_G, \theta_Q, A) - \hat{V}_m\}.$$

The function $Q(\theta)$ is non-negative for all $\theta$. $\theta^* \in \Theta^*$ if and only if $Q(\theta^*) = 0$, so the solutions in $\Theta^*$ can be characterized as

$$\Theta^* = \{\theta^* \in \Theta : Q(\theta^*) = 0\}.$$

Similarly, the solutions in $\hat{\Theta}_m(\tau_m)$ can be written as

$$\hat{\Theta}_m(\tau_m) = \{\hat{\theta} \in \Theta : \hat{Q}_m(\hat{\theta}) \leq \tau_m\}.$$

With this notation, we first show that

$$\sup_{\theta \in \Theta} |\hat{Q}_m(\theta) - Q(\theta)| \xrightarrow{p} 0 \quad \text{with the function } Q(\theta) \text{ continuous in } \theta. \tag{5.25}$$

We have

$$
\begin{aligned}
&|\hat{Q}_m(\theta) - Q(\theta)| \\
=&|\hat{\phi}_m(\theta_G, \theta_Q, A) - \phi(\theta_G, \theta_Q, A) - (\inf\{\hat{L}_m(\theta_G, \theta_Q, A, \theta_f), \hat{V}_m\} - \inf\{L(\theta_G, \theta_Q, A, \theta_f), V^*\})| \\
\leq&|\hat{\phi}_m(\theta_G, \theta_Q, A) - \phi(\theta_G, \theta_Q, A)| + \sup\{|\hat{L}_m(\theta) - L(\theta)|, |\hat{V}_m - V^*|\} \\
\leq&|\hat{\phi}_m(\theta_G, \theta_Q, A) - \phi(\theta_G, \theta_Q, A)| + \sup\{|\hat{L}_m(\theta) - L(\theta)|, \sup_{\theta_G, \theta_Q, A} |\hat{\phi}_m(\theta_G, \theta_Q, A) - \phi(\theta_G, \theta_Q, A)|\} \\
\leq&2 \sup_{\theta \in \Theta} |\hat{L}_m(\theta) - L(\theta)|
\end{aligned}
$$

The above inequalities result from the triangle inequality and elementary properties of min and max. Note that the assumption $\sup_{\theta\in\Theta} m^{1/2}|\hat{L}_m(\theta) - L(\theta)| = \mathcal{O}_p(1)$ implies that $\sup_{\theta\in\Theta} |\hat{L}_m(\theta) - L(\theta)| \xrightarrow{p} 0$, thus we have $\sup_{\theta\in\Theta} |\hat{Q}_m(\theta) - Q(\theta)| \xrightarrow{p} 0$. As for the continuity of $Q(\theta)$, the function $L(\theta)$ is continuous on the compact set $\Theta$, and thus by Berge's maximum theorem the function $\phi(\theta_G, \theta_Q, A) = \sup_{\theta_f} L(\theta_G, \theta_Q, A, \theta_f)$ is continuous on $(\theta_G, \theta_Q, A)$. Consequently, $Q(\theta)$ is continuous. The continuity of $Q$ and the definition of $\Theta^*$ imply that for all $\epsilon > 0$ there exists an $\eta(\epsilon) > 0$ such that

$$\inf_{\theta \in \Theta \setminus \Theta^*_\epsilon} Q(\theta) \geq \eta(\epsilon), \tag{5.26}$$

where $\Theta^*_\epsilon$ denotes the $\epsilon$-net of the set $\Theta^*$ in $\Theta$ defined as $\Theta^*_\epsilon := \{\theta \in \Theta : d(\theta, \Theta^*) \leq \epsilon\}$ and $\Theta \setminus \Theta^*_\epsilon$ is the complement of $\Theta^*_\epsilon$ in $\Theta$.

Now we are ready to show that $\sup_{\theta\in\hat{\Theta}_m(\tau_m)} d(\theta, \Theta^*) \xrightarrow{p} 0$. Let small $\epsilon_p, \epsilon_d > 0$ be arbitrary, choose an $\eta = \eta(\epsilon_d)$ such that $\inf_{\theta\in\Theta\setminus\Theta^*_{\epsilon_d}} Q(\theta) \geq \eta$ holds, and choose an $m_{\epsilon_p}$ such that for all $m \geq m_{\epsilon_p}$ both $\sup_{\theta\in\Theta} |\hat{Q}_m(\theta) - Q(\theta)| \leq \eta/4$ and $\tau_m \leq \eta/4$ hold with probability larger than $1 - \epsilon_p$. We now have

$$
\begin{aligned}
\sup_{\theta \in \hat{\Theta}_m(\tau_m)} Q(\theta) &\leq \sup_{\theta \in \hat{\Theta}_m(\tau_m)} |\hat{Q}_m(\theta) - Q(\theta)| + \sup_{\theta \in \hat{\Theta}_m(\tau_m)} \hat{Q}_m(\theta) \\
&\leq \sup_{\theta \in \Theta} |\hat{Q}_m(\theta) - Q(\theta)| + \tau_m \leq \eta/2 < \inf_{\theta \in \Theta \setminus \Theta^*_{\epsilon_d}} Q(\theta).
\end{aligned}
$$

Therefore $\hat{\Theta}_m(\tau_m) \subseteq \Theta^*_{\epsilon_d}$ and $\sup_{\theta\in\hat{\Theta}_m(\tau_m)} d(\theta,\Theta^*) \leq \epsilon_d$, for all $m \geq m_{\epsilon_p}$ with probability larger than $1-\epsilon_p$. Thus $\sup_{\theta\in\hat{\Theta}_m(\tau_m)} d(\theta,\Theta^*) \xrightarrow{p} 0$.

Then we prove $\sup_{\theta\in\Theta^*} d(\theta,\hat{\Theta}_m(\tau_m)) \xrightarrow{p} 0$. Using the assumption that $\sup_{\theta\in\Theta} m^{1/2}|\hat{L}_m(\theta)-L(\theta)| = \mathcal{O}_p(1)$, we have $\sup_{\theta\in\Theta} m^{1/2}|\hat{Q}_m(\theta)-Q(\theta)| = \mathcal{O}_p(1)$. Note that

$$\sup_{\theta\in\Theta^*} \hat{Q}_m(\theta) \leq \sup_{\theta\in\Theta^*} |\hat{Q}_m(\theta)-Q(\theta)| + \sup_{\theta\in\Theta^*} Q(\theta) \leq \sup_{\theta\in\Theta} |\hat{Q}_m(\theta)-Q(\theta)|,$$

as $\Theta^* \subseteq \Theta$ and $\sup_{\theta\in\Theta^*} Q(\theta) = 0$ by the definition of $\Theta^*$. By assumption $m^{-1/2}/\tau_m \xrightarrow{p} 0$, for any $\epsilon_p > 0$ we can find a $m_{\epsilon_p}$ such that for all $m \geq m_{\epsilon_p}$

$$\sup_{\theta\in\Theta^*} \hat{Q}_m(\theta) \leq O_p(m^{-1/2}) = O_p(1)(m^{-1/2}/\tau_m)\tau_m \leq \tau_m$$

with probability larger than $1-\epsilon_p$. By the definition of $\hat{\Theta}_m(\tau_m)$ we now have $\Theta^* \subseteq \hat{\Theta}_m(\tau_m)$ and thus $\sup_{\theta\in\Theta^*} d(\theta,\hat{\Theta}_m(\tau_m)) = 0$ for all $m \geq m_{\epsilon_p}$ with probability larger than $1-\epsilon_p$. This shows that $\sup_{\theta\in\Theta^*} d(\theta,\hat{\Theta}_m(\tau_m)) \xrightarrow{p} 0$. This completes the proof that $d_H(\hat{\Theta}_m(\tau_m),\Theta^*) \xrightarrow{p} 0$.

### 5.5.5 Proof of Theorem 5.3.2

**Lemma 5.5.1.** *Given a $L_G$-Lipschitz generator $G$, a set of 1-Lipschitz discriminators $\mathcal{F}$, and a set of decoders $\mathcal{Q}$ whose functions are $L_Q$-Lipschitz with respect to the input, and $L_{\theta_Q}$-Lipschitz with respect to its parameter $\theta_Q \in \Theta_Q$, let $\hat{\Theta}_Q$ be a $\epsilon/(16L_GL_{\theta_Q})$-net of the parameter space $\Theta_Q$ of $\mathcal{Q}$, and $\hat{\mathcal{A}}$ be a $\epsilon/(8\sqrt{5d}L_G)$-net of the parameter space $\mathcal{A}$ of $A$. Then the following inequality holds with a probability of at least*

$$1-\exp\{-d\} - 2|\hat{\Theta}_Q||\hat{\mathcal{A}}|\exp\{-\frac{\epsilon^2 m}{8(1+2L_GL_Q+\sqrt{5d}L_G\|A\|)^2}\}$$

*over the choice of m samples $S_x$ from the data distribution $P_X$ and m samples $S_{z_0}$ from the standard normal distribution $N(0, I)$.*

$$\sup_{A\in\mathcal{A},Q\in\mathcal{Q}}\Big|\mathbb{E}\|X-G(Q(X))\|+\mathbb{E}\sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x}f(G(Q(\mathbf{x})))-\hat{\mathbb{E}}_{S_{z_0}}f(G(A\mathbf{z}_0))\right\}$$
$$-\left\{\hat{\mathbb{E}}_{S_x}\|\mathbf{x}-G(Q(\mathbf{x}))\|+\sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x}f(G(Q(\mathbf{x})))-\hat{\mathbb{E}}_{S_{z_0}}f(G(A\mathbf{z}_0))\right\}\right\}\Big|\leq\epsilon$$

*where $|\hat{\Theta}_Q|$ denote the size of the finite set $\hat{\Theta}_Q$, and $|\hat{\mathcal{A}}|$ denote the size of the finite set $\hat{\mathcal{A}}$*

*Proof.* Since $\mathbf{z}_0\sim N(0,I_d)$, we have

$$\mathbb{P}(\|\mathbf{z}_0\|\leq\sqrt{5d})\geq 1-\exp(-d)$$

Then we have

$$\begin{aligned}
&\mathbb{P}\Big[\sup_{A\in\mathcal{A},Q\in\mathcal{Q}}\Big|\mathbb{E}\|X-G(Q(X))\|+\mathbb{E}\sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x}f(G(Q(\mathbf{x})))-\hat{\mathbb{E}}_{S_{z_0}}f(G(A\mathbf{z}_0))\right\}\\
&-\left\{\hat{\mathbb{E}}_{S_x}\|\mathbf{x}-G(Q(\mathbf{x}))\|+\sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x}f(G(Q(\mathbf{x})))-\hat{\mathbb{E}}_{S_{z_0}}f(G(A\mathbf{z}_0))\right\}\right\}\Big|\geq\epsilon\Big]\\
\leq&\mathbb{P}\Big[\|\mathbf{z}_0\|\leq\sqrt{5d},\sup_{A\in\mathcal{A},Q\in\mathcal{Q}}\Big|\mathbb{E}\|X-G(Q(X))\|+\mathbb{E}\sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x}f(G(Q(\mathbf{x})))-\hat{\mathbb{E}}_{S_{z_0}}f(G(A\mathbf{z}_0))\right\}\\
&-\left\{\hat{\mathbb{E}}_{S_x}\|\mathbf{x}-G(Q(\mathbf{x}))\|+\sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x}f(G(Q(\mathbf{x})))-\hat{\mathbb{E}}_{S_{z_0}}f(G(A\mathbf{z}_0))\right\}\right\}\Big|\geq\epsilon\Big]\\
&+\mathbb{P}\Big[\|\mathbf{z}_0\|>\sqrt{5d}\Big].
\end{aligned}$$

Let us first prove the former part. Without loss of generality, we combine two sets $S_x$ and $S_{z_0}$ together, and write it as $S=\{(\mathbf{x}_1,\mathbf{z}_{0,1}),\ldots,(\mathbf{x}_m,\mathbf{z}_{0,m})\}$ For convenience, we define

$$\Psi((\mathbf{x}_1,\mathbf{z}_{0,1}),\ldots,(\mathbf{x}_m,\mathbf{z}_{0,m}))=\hat{\mathbb{E}}_{S_x}\|\mathbf{x}-G(Q(\mathbf{x}))\|+\sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x}f(G(Q(\mathbf{x})))-\hat{\mathbb{E}}_{S_{z_0}}f(G(A\mathbf{z}_0))\right\}$$

Suppose that there is another sample set $S' = \{(\mathbf{x}_1, z_{0,1}), \ldots, (\mathbf{x}'_{\mathrm{i}}, \mathbf{z}'_{0,\mathrm{i}}), \ldots, (\mathbf{x}_m, z_{0,m})\}$, which differs from $S$ by exactly one element. Then it is clear that

$$\begin{aligned}\left|\hat{\mathbb{E}}_{S_x}\|\mathbf{x} - G(Q(\mathbf{x}))\| - \hat{\mathbb{E}}_{S'_x}\|\mathbf{x} - G(Q(\mathbf{x}))\|\right| =& \left|\frac{1}{m}\|\mathbf{x}_{\mathrm{i}} - G(Q(\mathbf{x}_{\mathrm{i}}))\| - \frac{1}{m}\|\mathbf{x}'_{\mathrm{i}} - G(Q(\mathbf{x}'_{\mathrm{i}}))\|\right| \\ \leq& \frac{\|\mathbf{x}_{\mathrm{i}} - \mathbf{x}'_{\mathrm{i}}\| + \|G(Q(\mathbf{x}_{\mathrm{i}})) - G(Q(\mathbf{x}'_{\mathrm{i}}))\|}{m} \\ \leq& \frac{2(1 + L_G L_Q)}{m},\end{aligned}$$

where the last inequality is due to to the Lipschitz continuity of $f$, $G$ and $Q$. And

$$\begin{aligned}&\left|\sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x} f(G(Q(\mathbf{x}))) - \hat{\mathbb{E}}_{S_{z_0}} f(G(A\mathbf{z}_0))\right\} - \sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S'_x} f(G(Q(\mathbf{x}))) - \hat{\mathbb{E}}_{S'_{z_0}} f(G(A\mathbf{z}_0))\right\}\right| \\ \leq& \sup_{f\in\mathcal{F}}\left|\hat{\mathbb{E}}_{S_x} f(G(Q(\mathbf{x}))) - \hat{\mathbb{E}}_{S'_x} f(G(Q(\mathbf{x})))\right| + \sup_{f\in\mathcal{F}}\left|\hat{\mathbb{E}}_{S_{z_0}} f(G(A\mathbf{z}_0)) - \hat{\mathbb{E}}_{S'_{z_0}} f(G(A\mathbf{z}_0))\right| \\ =& \frac{1}{m}\sup_{f\in\mathcal{F}}\left|f(G(Q(\mathbf{x}_{\mathrm{i}}))) - f(G(Q(\mathbf{x}'_{\mathrm{i}})))\right| + \frac{1}{m}\sup_{f\in\mathcal{F}}\left|f(G(A\mathbf{z}_{0,\mathrm{i}})) - f(G(A\mathbf{z}'_{0,\mathrm{i}}))\right| \\ \leq& \frac{2L_G(L_Q + \sqrt{5d}\|A\|)}{m},\end{aligned}$$

Therefore,

$$\begin{aligned}&\left|\Psi((\mathbf{x}_1, \mathbf{z}_{0,1}), \ldots, (\mathbf{x}_{\mathrm{i}}, \mathbf{z}_{0,1}), \ldots, (\mathbf{x}_m, \mathbf{z}_{0,m})) - \Psi((\mathbf{x}_1, \mathbf{z}_{0,1}), \ldots, (\mathbf{x}'_{\mathrm{i}}, \mathbf{z}'_{0,1}), \ldots, (\mathbf{x}_m, \mathbf{z}_{0,m}))\right| \\ \leq& \frac{2(1 + 2L_G L_Q + \sqrt{5d}L_G\|A\|)}{m}\end{aligned}$$

Applying McDiamond's inequality, it holds that

$$\begin{aligned}&\mathbb{P}\left[\left|\Psi((\mathbf{x}_1, \mathbf{z}_{0,1}), \ldots, (\mathbf{x}_m, \mathbf{z}_{0,m})) - \mathbb{E}\Psi((\mathbf{x}_1, \mathbf{z}_{0,1}), \ldots, (\mathbf{x}_m, \mathbf{z}_{0,m}))\right| \geq \frac{\epsilon}{2}\right] \\ \leq& 2\exp\left\{-\frac{\epsilon^2 m}{8(1 + 2L_G L_Q + \sqrt{5d}L_G\|A\|)^2}\right\}\end{aligned}$$

Then by a union bound over all $\mathcal{Q}_{\hat{\Theta}_Q}$ and $\hat{\mathcal{A}}$, we have

$$\begin{aligned}
&\mathbb{P}\Big[\sup_{A\in\hat{\mathcal{A}},Q\in\mathcal{Q}_{\hat{\Theta}_Q}}\Big|\Psi((\mathbf{x}_1,\mathbf{z}_{0,1}),\ldots,(\mathbf{x}_m,\mathbf{z}_{0,m}))-\mathbb{E}\Psi((\mathbf{x}_1,\mathbf{z}_{0,1}),\ldots,(\mathbf{x}_m,\mathbf{z}_{0,m}))\Big|\geq\frac{\epsilon}{2}\Big]\\
\leq&2|\hat{\Theta}_Q||\hat{\mathcal{A}}|\exp\{-\frac{\epsilon^2 m}{8(1+2L_GL_Q+\sqrt{5d}L_G\|A\|)^2}\}.
\end{aligned}$$

Since $\hat{\Theta}_Q$ is a $\epsilon/(16L_GL_{\theta_Q})$-net of the parameter space $\Theta_Q$ of $\mathcal{Q}$, every point in $\Theta_Q$ is within distance $\epsilon/(16L_GL_{\theta_Q})$ of a point in $\hat{\Theta}_Q$. Similarly, every point in $\mathcal{A}$ is within distance $\epsilon/(8\sqrt{5d}L_G)$ of a point in $\hat{\mathcal{A}}$. For any $Q\in\mathcal{Q}$ and $A\in\mathcal{A}$, there exists a $Q'\in\mathcal{Q}_{\hat{\Theta}_Q}$ and $A'\in\hat{\mathcal{A}}$, such that

$$\begin{aligned}
&\Big|\mathbb{E}\|X-G(Q(X))\|+\mathbb{E}\sup_{f\in\mathcal{F}}\big\{\hat{\mathbb{E}}_{S_x}f(G(Q(\mathbf{x})))-\hat{\mathbb{E}}_{S_{z_0}}f(G(A\mathbf{z}_0))\big\}\\
&-\Big\{\mathbb{E}\|X-G(Q'(X))\|+\mathbb{E}\sup_{f\in\mathcal{F}}\big\{\hat{\mathbb{E}}_{S_x}f(G(Q'(X)))-\hat{\mathbb{E}}_{S_{z_0}}f(G(A'Z_0))\big\}\Big\}\Big|\\
\leq&\Big|\mathbb{E}\|X-G(Q(X))\|-\mathbb{E}\|X-G(Q'(X))\|\Big|\\
&+\Big|\mathbb{E}\sup_{f\in\mathcal{F}}\big\{\hat{\mathbb{E}}_{S_x}f(G(Q(\mathbf{x})))-\hat{\mathbb{E}}_{S_{z_0}}f(G(A\mathbf{z}_0))\big\}-\mathbb{E}\sup_{f\in\mathcal{F}}\big\{\hat{\mathbb{E}}_{S_x}f(G(Q'(\mathbf{x})))-\hat{\mathbb{E}}_{S_{z_0}}f(G(A'\mathbf{z}_0))\big\}\Big|\\
\leq&L_GL_{\theta_Q}\frac{\epsilon}{16L_GL_{\theta_Q}}\\
&+\mathbb{E}\sup_{f\in\mathcal{F}}\Big\{\Big|\hat{\mathbb{E}}_{S_x}f(G(Q(\mathbf{x})))-\hat{\mathbb{E}}_{S_x}f(G(Q'(\mathbf{x})))\Big|+\Big|\hat{\mathbb{E}}_{S_{z_0}}f(G(A\mathbf{z}_0))-\hat{\mathbb{E}}_{S_{z_0}}f(G(A'\mathbf{z}_0))\Big|\Big\}\\
\leq&\frac{\epsilon}{16}+L_GL_{\theta_Q}\frac{\epsilon}{16L_GL_{\theta_Q}}+\sqrt{5d}L_G\frac{\epsilon}{8\sqrt{5d}L_G}=\frac{\epsilon}{4}
\end{aligned}$$

Using the same strategy, we have

$$\begin{aligned}
&\Big|\hat{\mathbb{E}}\|\mathbf{x}-G(Q'(\mathbf{x}))\|+\sup_{f\in\mathcal{F}}\big\{\hat{\mathbb{E}}_{S_x}f(G(Q'(\mathbf{x})))-\hat{\mathbb{E}}_{S_{z_0}}f(G(A'\mathbf{z}_0))\big\}\\
&-\Big\{\hat{\mathbb{E}}\|\mathbf{x}-G(Q(\mathbf{x}))\|+\sup_{f\in\mathcal{F}}\big\{\hat{\mathbb{E}}_{S_x}f(G(Q(\mathbf{x})))-\hat{\mathbb{E}}_{S_{z_0}}f(G(A\mathbf{z}_0))\big\}\Big\}\Big|\leq\frac{\epsilon}{4}
\end{aligned}$$

Therefore,

$$
\begin{aligned}
&\Big|\mathbb{E}\|X - G(Q(X))\| + \mathbb{E}\sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x} f(G(Q(\mathbf{x}))) - \hat{\mathbb{E}}_{S_{z_0}} f(G(A\mathbf{z}_0))\right\} \\
&\quad - \left\{\hat{\mathbb{E}}_{S_x}\|\mathbf{x} - G(Q(\mathbf{x}))\| + \sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x} f(G(Q(\mathbf{x}))) - \hat{\mathbb{E}}_{S_{z_0}} f(G(A\mathbf{z}_0))\right\}\right\}\Big| \\
\leq&\Big|\mathbb{E}\|X - G(Q(X))\| + \mathbb{E}\sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x} f(G(Q(\mathbf{x}))) - \hat{\mathbb{E}}_{S_{z_0}} f(G(A\mathbf{z}_0))\right\} \\
&\quad - \left\{\mathbb{E}\|X - G(Q'(X))\| + \mathbb{E}\sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x} f(G(Q'(X))) - \hat{\mathbb{E}}_{S_{z_0}} f(G(A'Z_0))\right\}\right\}\Big| \\
&\quad + \Big|\mathbb{E}\|X - G(Q'(X))\| + \mathbb{E}\sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x} f(G(Q'(X))) - \hat{\mathbb{E}}_{S_{z_0}} f(G(A'Z_0))\right\} \\
&\quad - \left\{\hat{\mathbb{E}}_{S_x}\|\mathbf{x} - G(Q'(\mathbf{x}))\| + \sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x} f(G(Q'(\mathbf{x}))) - \hat{\mathbb{E}}_{S_{z_0}} f(G(A'\mathbf{z}_0))\right\}\right\}\Big| \\
&\quad + \Big|\hat{\mathbb{E}}_{S_x}\|\mathbf{x} - G(Q'(\mathbf{x}))\| + \sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x} f(G(Q'(\mathbf{x}))) - \hat{\mathbb{E}}_{S_{z_0}} f(G(A'\mathbf{z}_0))\right\} \\
&\quad - \left\{\hat{\mathbb{E}}_{S_x}\|\mathbf{x} - G(Q(\mathbf{x}))\| + \sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x} f(G(Q(\mathbf{x}))) - \hat{\mathbb{E}}_{S_{z_0}} f(G(A\mathbf{z}_0))\right\}\right\}\Big| \\
\leq&\frac{\epsilon}{4} + \frac{\epsilon}{2} + \frac{\epsilon}{4} = \epsilon
\end{aligned}
$$

Finally, combining with $\mathbb{P}[\|\mathbf{z}_0\| > \sqrt{5d}] \leq \exp(-d)$, we have

$$
\begin{aligned}
&\sup_{A\in\mathcal{A},Q\in\mathcal{Q}}\Big|\mathbb{E}\|X - G(Q(X))\| + \mathbb{E}\sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x} f(G(Q(\mathbf{x}))) - \hat{\mathbb{E}}_{S_{z_0}} f(G(A\mathbf{z}_0))\right\} \\
&- \left\{\hat{\mathbb{E}}_{S_x}\|\mathbf{x} - G(Q(\mathbf{x}))\| + \sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x} f(G(Q(\mathbf{x}))) - \hat{\mathbb{E}}_{S_{z_0}} f(G(A\mathbf{z}_0))\right\}\right\}\Big| \leq \epsilon
\end{aligned}
$$

holding with probability at least $1 - 2|\hat{\Theta}_Q||\hat{\mathcal{A}}|\exp\{-\frac{\epsilon^2 m}{8(1+2L_G L_Q+\sqrt{5d}L_G\|A\|)^2}\} - \exp\{-d\}$. □

**Lemma 5.5.2.** *Given the function class $\mathcal{F}$, $\mathcal{Q}$, and a set of parameter $\mathcal{A}$,*

$$
\begin{aligned}
&\sup_{A\in\mathcal{A},Q\in\mathcal{Q}}\Big|\sup_{f\in\mathcal{F}}\{\mathbb{E}f(G(Q(X))) - \mathbb{E}f(G(AZ_0))\} - \mathbb{E}\sup_{f\in\mathcal{F}}\{\hat{\mathbb{E}}_{S_x} f(G(Q(\mathbf{x}))) - \hat{\mathbb{E}}_{S_{z_0}} f(G(A\mathbf{z}_0))\}\Big| \\
\leq&2\mathfrak{R}_m(\mathcal{F}\circ G\circ\mathcal{Q}) + 2\mathfrak{R}_m(\mathcal{F}\circ G\circ\mathcal{A}).
\end{aligned}
$$

*Proof.*

$$
\begin{aligned}
&\sup_{A\in\mathcal{A},Q\in\mathcal{Q}}\left|\sup_{f\in\mathcal{F}}\left\{\mathbb{E}f(G(Q(X)))-\mathbb{E}f(G(AZ_0))\right\}-\mathbb{E}\sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x}f(G(Q(\mathbf{x})))-\hat{\mathbb{E}}_{S_{z_0}}f(G(A\mathbf{z}_0))\right\}\right|\\
\leq&\sup_{A\in\mathcal{A},Q\in\mathcal{Q}}\mathbb{E}\left|\sup_{f\in\mathcal{F}}\left\{\mathbb{E}f(G(Q(X)))-\mathbb{E}f(G(AZ_0))\right\}-\sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x}f(G(Q(\mathbf{x})))-\hat{\mathbb{E}}_{S_{z_0}}f(G(A\mathbf{z}_0))\right\}\right|\\
\leq&\sup_{A\in\mathcal{A},Q\in\mathcal{Q}}\mathbb{E}\sup_{f\in\mathcal{F}}\left|\left[\mathbb{E}f(G(Q(X)))-\hat{\mathbb{E}}_{S_x}f(G(Q(\mathbf{x})))\right]+\left[\hat{\mathbb{E}}_{S_{z_0}}f(G(A\mathbf{z}_0))-\mathbb{E}f(G(AZ_0))\right]\right|\\
\leq&\mathbb{E}\sup_{A\in\mathcal{A},Q\in\mathcal{Q},f\in\mathcal{F}}\left|\left[\mathbb{E}f(G(Q(X)))-\hat{\mathbb{E}}_{S_x}f(G(Q(\mathbf{x})))\right]+\left[\hat{\mathbb{E}}_{S_{z_0}}f(G(A\mathbf{z}_0))-\mathbb{E}f(G(AZ_0))\right]\right|\\
\leq&\mathbb{E}\sup_{Q\in\mathcal{Q},f\in\mathcal{F}}\left|\mathbb{E}f(G(Q(X)))-\hat{\mathbb{E}}_{S_x}f(G(Q(\mathbf{x})))\right|+\mathbb{E}\sup_{A\in\mathcal{A},f\in\mathcal{F}}\left|\mathbb{E}f(G(AZ_0))-\hat{\mathbb{E}}_{S_{z_0}}f(G(A\mathbf{z}_0))\right|\\
\leq&2\mathfrak{R}_m(\mathcal{F}\circ G\circ\mathcal{Q})+2\mathfrak{R}_m(\mathcal{F}\circ G\circ\mathcal{A}),
\end{aligned}
$$

The last inequality is obtained by the standard technique of symmetrization in [75]. □

Equipped with the above two lemmas, we are now able to prove the main theorem. Firstly, we have

$$
\begin{aligned}
&\sup_{A\in\mathcal{A}}\left|\overline{W}_1(P_X,P_{G(Z^*)})-\overline{W}_1(\hat{P}_X,\hat{P}_{G(Z^*)})\right|\\
=&\sup_{A\in\mathcal{A}}\left|\inf_{Q\in\mathcal{Q}}\sup_{f\in\mathcal{F}}\left\{\mathbb{E}\|X-G(Q(X))\|+\mathbb{E}f(G(Q(X)))-\mathbb{E}f(G(AZ_0))\right\}\right.\\
&\left.-\inf_{Q\in\mathcal{Q}}\sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x}\|\mathbf{x}-G(Q(\mathbf{x}))\|+\hat{\mathbb{E}}_{S_x}f(G(Q(\mathbf{x})))-\hat{\mathbb{E}}_{S_{z_0}}f(G(A\mathbf{z}_0)\right\}\right|\\
\leq&\sup_{A\in\mathcal{A},Q\in\mathcal{Q}}\left|\mathbb{E}\|X-G(Q(X))\|-\hat{\mathbb{E}}_{S_x}\|\mathbf{x}-G(Q(\mathbf{x}))\|+\sup_{f\in\mathcal{F}}\left\{\mathbb{E}f(G(Q(X)))-\mathbb{E}f(G(AZ_0))\right\}\right.\\
&\left.-\sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x}f(G(Q(\mathbf{x})))-\hat{\mathbb{E}}_{S_{z_0}}f(G(A\mathbf{z}_0))\right\}\right|\\
\leq&\sup_{A\in\mathcal{A},Q\in\mathcal{Q}}\left|\mathbb{E}\|X-G(Q(X))\|+\mathbb{E}\sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x}f(G(Q(\mathbf{x})))-\hat{\mathbb{E}}_{S_{z_0}}f(G(A\mathbf{z}_0))\right\}\right.\\
&\left.-\hat{\mathbb{E}}_{S_x}\|\mathbf{x}-G(Q(\mathbf{x}))\|-\sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x}f(G(Q(\mathbf{x})))-\hat{\mathbb{E}}_{S_{z_0}}f(G(A\mathbf{z}_0))\right\}\right|\\
&+\sup_{A\in\mathcal{A},Q\in\mathcal{Q}}\left|\sup_{f\in\mathcal{F}}\left\{\mathbb{E}f(G(Q(X)))-\mathbb{E}f(G(AZ_0))\right\}-\mathbb{E}\sup_{f\in\mathcal{F}}\left\{\hat{\mathbb{E}}_{S_x}f(G(Q(\mathbf{x})))-\hat{\mathbb{E}}_{S_{z_0}}f(G(A\mathbf{z}_0))\right\}\right|\\
\leq&2\mathfrak{R}_m(\mathcal{F}\circ G\circ\mathcal{Q})+2\mathfrak{R}_m(\mathcal{F}\circ G\circ\mathcal{A})+\epsilon
\end{aligned}
$$

with probability at least $1-2|\hat{\Theta}_Q||\hat{\mathcal{A}}|\exp\{-\frac{\epsilon^2 m}{8(1+2L_GL_Q+\sqrt{5d}L_G\|A\|)^2}\}-\exp\{-d\}$.

# 6. CONCLUSION

As a summary, in the first part of the dissertation, we study the generalization behavior for robust adversarial learning. We focus on the $\ell_\infty$ adversarial attacks and analyze generalization through the lens of Rademacher complexity. In particular, we view deep neural networks as a composition of a shallower network and a Lipschitz continuous function on a low dimension and study the weight normalization based on both the spectral norm and the rank constraints. We establish tight complexity bounds for adversarial learning and realize that the effect of adversarial perturbations can be limited under this weight normalization.

Several future directions for research will be pursued. Our theoretical establishment implies that high-probability learning can be guaranteed for algorithms which provide predictors within this class. We are investigating efficient and practical algorithms in the context of norm and rank based constraints. One possible algorithm is to combine the spectral normalization in [52] with the dropout. Much of this would be experimental. Another theoretical problem is to develop generalization bounds for more sophisticated networks such as CNNs. One remaining theoretical problem is to develop the sharp lower bound for the adversarial Rademacher complexity for neural works.

In the second part of this dissertation, we have developed a novel LWGAN framework that enables us to adaptively learn the intrinsic dimension of the data distribution. This framework fuses the WAE and the WGAN in a natural primal and dual way, so that the encoder learns a latent normal distribution whose rank of the covariance matrix is exactly equivalent to the dimension of the data manifold. We have provide the estimation consistency and an upper bound on the generalization error. Our algorithm have shown that the intrinsic dimension of the data can be successfully detected under several settings on both synthetic dataset and benchmark dataset. The empirical results have showed that the generative data by the LWGAN is high-quality.

In the future direction of research on LWGAN, we will investigate a more general scenario with a stochastic generator $G$ as illustrate in Corollary 5.1.1. Using the noise-outscoring lemma, theoretically only an extra noise $\eta \sim N(0,1)$ is needed to be added to the input of the generator $G$. Practically, we may need to use a higher dimensional noise vector to ease

the representation of $G$. In addition, it is interesting to incorporate the stochastic LWGAN into the recent GAN moduls such as BigGAN [76] so that high-resolution and high-fidelity images can be simultaneously produced when the intrinsic dimension is detected.

The new LWGAN framework has many potential applications in other fields. For example, the LWGAN can be used for structural estimation. Structural estimation is a useful tool to quantify economic mechanisms and learn about the effects of policies that are yet to be implemented **wei20**. An economic structural model specifies some outcome $g(x, \epsilon; \theta)$ that depends on a set of observables $x$, unobservables $\epsilon$, and structural parameters $\theta$. The function $g$ can represent a utility maximization problem or other observed outcomes. Under many scenarios, the likelihood function and moment functions are not easy to obtain. This makes the MLE and GMM infeasible, and other simulation based methods can cause additional computational burden. By the training of the LWGAN on the data from $(x, y)$, we are able to adaptively learning the data representation by the encoder, instead of using moments. We are also able to boost the sample size by the generator. By comparing the generated data $(x, g(x, \epsilon; \theta))$ and the observed data $(x, y)$ in the latent space, we can estimate $\theta$ efficiently.

# REFERENCES


[1] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[2] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," vol. 84, 2017.

[3] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma, "Adversarial classification," 2004.

[4] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "There is no free lunch in adversarial robustness (but there are unexpected benefits)," 2018.

[5] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *Computer Science*, 2013.

[6] L. Engstrom, D. Tsipras, L. Schmidt, and A. Madry, "A rotation and a translation suffice: Fooling cnns with simple transformations," 2017.

[7] J. Gilmer, R. P. Adams, I. Goodfellow, D. Andersen, and G. E. Dahl, "Motivating the rules of the game for adversarial example research," 2018.

[8] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017.

[9] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *Eprint Arxiv*, 2014.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," 2012, pp. 1097–1105.

[11] O. Schwartz, "You thought fake news was bad? deep fakes are where truth goes to die," *The Guardian*, 2018.

[12] A. Raghunathan, J. Steinhardt, and P. Liang, "Certified defenses against adversarial examples," *arXiv preprint arXiv:1801.09344*, 2018.

[13] K. Y. Xiao, V. Tjeng, N. M. Shafiullah, and A. Madry, "Training for faster adversarial robustness verification via inducing relu stability," *arXiv preprint arXiv:1809.03008*, 2018.

[14] J. Z. Kolter and E. Wong, "Provable defenses against adversarial examples via the convex outer adversarial polytope," 2018.

[15] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, and M. I. Jordan, "Theoretically principled trade-off between robustness and accuracy," 2019.

[16] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *Computer Science*, 2014.

[17] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Mądry, "Adversarially robust generalization requires more data," 2018.

[18] F. Farnia, J. M. Zhang, and D. Tse, "Generalizable adversarial training via spectral normalization," *arXiv preprint arXiv:1811.07457*, 2018.

[19] D. Yin, K. Ramchandran, and P. Bartlett, "Rademacher complexity for adversarially robust generalization," *arXiv preprint arXiv:1810.11914*, 2018.

[20] J. Khim and P.-L. Loh, "Adversarial risk bounds for binary classification via function transformation," *arXiv preprint arXiv:1810.09519*, 2018.

[21] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

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

[23] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*, PMLR, 2017, pp. 214–223.

[24] Y. Li, K. Swersky, and R. Zemel, "Generative moment matching networks," in *International Conference on Machine Learning*, PMLR, 2015, pp. 1718–1727.

[25] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," *arXiv preprint arXiv:1605.08803*, 2016.

[26] Y. Qiu and X. Wang, "Almond: Adaptive latent modeling and optimization via neural networks and langevin diffusion," *Journal of the American Statistical Association*, pp. 1–13, 2019.

[27] P. K. Rubenstein, B. Schoelkopf, and I. Tolstikhin, "On the latent space of wasserstein auto-encoders," *arXiv preprint arXiv:1802.03761*, 2018.

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

[29] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville, "Adversarially learned inference," in *International Conference on Learning Representations (ICLR)*, 2017.

[30] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," in *International Conference on Learning Representations (ICLR)*, 2017.

[31] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[32] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," 1992, pp. 950–957.

[33] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[34] J. Ba and B. Frey, "Adaptive dropout for training deep neural networks," 2013, pp. 3084–3092.

[35] Y. Yao, L. Rosasco, and A. Caponnetto, "On early stopping in gradient descent learning," *Constructive Approximation*, vol. 26, no. 2, pp. 289–315, 2007.

[36] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, 2017, pp. 5767–5777.

[37] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf, "Wasserstein auto-encoders," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=HkL7n1-0b.

[38] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, "Improved variational inference with inverse autoregressive flow," *Advances in neural information processing systems*, vol. 29, pp. 4743–4751, 2016.

[39] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," *arXiv preprint arXiv:1807.03039*, 2018.

[40] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning:* 2012.

[41] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.

[42] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[43] P. L. Bartlett, "The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network," *IEEE transactions on Information Theory*, vol. 44, no. 2, pp. 525–536, 1998.

[44] B. Neyshabur, R. Tomioka, and N. Srebro, "Norm-based capacity control in neural networks," in *Conference on Learning Theory*, 2015, pp. 1376–1401.

[45] B. Neyshabur, S. Bhojanapalli, and N. Srebro, "A PAC-bayesian approach to spectrally-normalized margin bounds for neural networks," 2018.

[46] S. Sun, W. Chen, L. Wang, X. Liu, and T.-Y. Liu, "On the depth of deep neural networks: A theoretical view," 2016, pp. 2066–2072.

[47] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky, "Spectrally-normalized margin bounds for neural networks," 2017, pp. 6240–6249.

[48] N. Golowich, A. Rakhlin, and O. Shamir, "Size-independent sample complexity of neural networks," *arXiv preprint arXiv:1712.06541*, 2017.

[49] X. Li, J. Lu, Z. Wang, J. Haupt, and T. Zhao, "On tighter generalization bound for deep neural networks: Cnns, resnets, and beyond," *arXiv preprint arXiv:1806.05159*, 2018.

[50] Y. Xu and X. Wang, "Understanding weight normalized deep neural networks with rectified linear units," 2018, pp. 130–139.

[51] Y. Yoshida and T. Miyato, "Spectral norm regularization for improving the generalizability of deep learning," *arXiv preprint arXiv:1705.10941*, 2017.

[52] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," 2018.

[53] J. Cavazza, P. Morerio, B. Haeffele, C. Lane, V. Murino, and R. Vidal, "Dropout as a low-rank regularizer for matrix factorization," *arXiv preprint arXiv:arXiv:1710.05092*, 2017.

[54] P. Mianjy, R. Arora, and R. Vidal, "On the implicit bias of dropout," *arXiv preprint arXiv:arXiv:1806.09777*, 2018.

[55] U. v. Luxburg and O. Bousquet, "Distance-based classification with lipschitz functions," *Journal of Machine Learning Research*, vol. 5, no. Jun, pp. 669–695, 2004.

[56] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

[57] E. J. Candes and Y. Plan, "Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements," *IEEE Transactions on Information Theory*, vol. 57, no. 4, pp. 2342–2359, 2011.

[58] E. J. Candes and Y. Plan, "Tight oracle bounds for low-rank matrix recovery from a minimal number of random measurements," *Mathematics*, vol. 57, no. 4, pp. 2342–2359, 2010.

[59] H. Whitney, J. Eells, and D. Toledo, *Collected Papers of Hassler Whitney*. Nelson Thornes, 1992, vol. 1.

[60] J. M. Lee, "Smooth manifolds," in *Introduction to Smooth Manifolds*, Springer, 2013, pp. 1–31.

[61] O. Kallenberg and O. Kallenberg, *Foundations of modern probability*. Springer, 1997, vol. 2.

[62] T. Austin, "Exchangeable random measures," in *Annales de l'IHP Probabilités et statistiques*, vol. 51, 2015, pp. 842–861.

[63] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural networks*, vol. 4, no. 2, pp. 251–257, 1991.

[64] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, no. Mar, pp. 723–773, 2012.

[65] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference for Learning Representations, San Diego, 2015*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980.

[66] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*. Springer Science & Business Media, 2009, vol. 317.

[67] M. Meitz, "Statistical inference for generative adversarial networks," *arXiv preprint arXiv:2104.10601*, 2021.

[68] J. Hoffmann-Jørgensen, *Probability with a view towards statistics*. Chapman and Hall, New York., 1994, vol. 2.

[69] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang, "Generalization and equilibrium in generative adversarial nets (gans)," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR. org, 2017, pp. 224–232.

[70] Q. Gao and X. Wang, "Theoretical investigation of generalization bounds for adversarial learning of deep neural networks," *Journal of Statistical Theory and Practice*, vol. 15, no. 2, pp. 1–28, 2021.

[71] J. A. Costa and A. O. Hero, "Determining intrinsic dimension and entropy of high-dimensional shape spaces," in *Statistics and Analysis of Shapes*, Springer, 2006, pp. 231–252.

[72] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in neural information processing systems*, 2016, pp. 2234–2242.

[73] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems*, 2017, pp. 6626–6637.

[74] M. Rosenblatt, "Remarks on a multivariate transformation," *Annals of Mathematical Statistics*, vol. 23, no. 3, pp. 470–472, 1952.

[75] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2018.

[76] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=B1xsqj09Fm.

# A. MODEL ARCHITECTURES OF SECTION 5.4

In this section, we present the architectures used for each experiment.

## A.1 Toy Examples

For Swiss Roll, S-curve, Hyperplane and Mixture Guassians, the latent space $Z \in \mathbb{R}^5$ and $Z \in \mathbb{R}^10$, for each batch, the sample size is 256.

Encoder architecture:

$$\begin{aligned} x \in \mathbb{R}^{dim_x} &\rightarrow FC_{1024} \rightarrow RELU \\ &\rightarrow FC_{512} \rightarrow RELU \\ &\rightarrow FC_{256} \rightarrow RELU \\ &\rightarrow FC_{128} \rightarrow RELU \rightarrow FC_5(FC_{10}) \end{aligned}$$

Generator architecture:

$$\begin{aligned} z \in \mathbb{R}^5(\mathbb{R}^{10}) &\rightarrow FC_{512} \rightarrow RELU \\ &\rightarrow FC_{512} \rightarrow RELU \\ &\rightarrow FC_{512} \rightarrow RELU \rightarrow FC_{dim_x} \end{aligned}$$

Discriminator architecture:

$$\begin{aligned} x \in \mathbb{R}^{dim_x} &\rightarrow FC_{512} \rightarrow RELU \\ &\rightarrow FC_{512} \rightarrow RELU \\ &\rightarrow FC_{512} \rightarrow RELU \rightarrow FC_1 \end{aligned}$$

## A.2 MNIST

For MNIST, the latent space $Z \in \mathbb{R}^{64}$ and batch size is 256.

Encoder architecture:

$$\begin{aligned} x \in \mathbb{R}^{28\times 28} &\rightarrow Conv_{128} \rightarrow RELU \\ &\rightarrow Conv_{256} \rightarrow RELU \\ &\rightarrow Conv_{512} \rightarrow RELU \rightarrow FC_{64} \end{aligned}$$

Generator architecture:

$$\begin{aligned} z \in \mathbb{R}^{64} &\rightarrow FC_{4\times 4\times 512} \rightarrow RELU \\ &\rightarrow ConvTrans_{256} \rightarrow RELU \\ &\rightarrow ConvTrans_{128} \rightarrow RELU \rightarrow ConvTrans_{1} \end{aligned}$$

Discriminator architecture:

$$\begin{aligned} x \in \mathbb{R}^{28\times 28} &\rightarrow Conv_{128} \rightarrow RELU \\ &\rightarrow Conv_{256} \rightarrow RELU \\ &\rightarrow Conv_{512} \rightarrow RELU \rightarrow FC_{1} \end{aligned}$$

## A.3 CelebA

For CelebA, the latent space $Z \in \mathbb{R}^{128}$ and batch size is 128.

Encoder architecture:

$$\begin{aligned} x \in \mathbb{R}^{64\times64\times3} &\rightarrow Conv_{128} \rightarrow LeakyRELU \\ &\rightarrow Conv_{256} \rightarrow InstanceNorm \rightarrow LeakyRELU \\ &\rightarrow Conv_{512} \rightarrow InstanceNorm \rightarrow LeakyRELU \\ &\rightarrow FC_{4\times4\times1024} \rightarrow FC_{128} \end{aligned}$$

Generator architecture:

$$\begin{aligned} z \in \mathbb{R}^{128} &\rightarrow FC_{4\times4\times1024} \\ &\rightarrow ConvTrans_{512} \rightarrow BN \rightarrow RELU \\ &\rightarrow ConvTrans_{256} \rightarrow BN \rightarrow RELU \\ &\rightarrow ConvTrans_{128} \rightarrow BN \rightarrow RELU \rightarrow ConvTrans_{3} \end{aligned}$$

Discriminator architecture:

$$\begin{aligned} x \in \mathbb{R}^{64\times64\times3} &\rightarrow Conv_{128} \rightarrow LeakyRELU \\ &\rightarrow Conv_{256} \rightarrow InstanceNorm \rightarrow LeakyRELU \\ &\rightarrow Conv_{512} \rightarrow InstanceNorm \rightarrow LeakyRELU \rightarrow Conv_{1} \end{aligned}$$

# VITA

Qingyi Gao was born in 1992 in Yunnan, China. She obtained a B.S. degree in Statistics from the School of Mathematical Sciences and a M.S. degree in Statistics from School of Statistics and Data Science at Nankai University. After then, she joined Department of statistics at Purdue University in January 2017 and earned a Ph.D. degree in Statistics in August 2021. Qingyi's research interests include statistical machine learning, learning theory, adversarial learning and deep generative models. After graduation, Qingyi would join the Facebook as a research data scientist.

# PUBLICATIONS AND PREPRINTS

**Publications:**

- **Gao, Q.**, Wang, X. (2021). Theoretical Investigation of Generalization Bounds for Adversarial Learning of Deep Neural Networks. *Journal of Statistical Theory and Practice.* DOI: 10.1007/s42519-021-00171-6.

- **Gao, Q.**, Wang, X. (2020). Statistical Learning. *Springer Handbook of Engineering Statistics, 2nd ed.* In press.

- Chen, Y., **Gao, Q.**, Liang, F., Wang, X. (2020). Nonlinear Variable Selection via Deep Neural Networks. *Journal of Computational and Graphical Statistics.* DOI: 10.1080/10618600.2020.1814305.

- Cai, J., **Gao, Q.**, Chun, H., Cai, H., Nantung, T. (2019). Spatial Autocorrelation in Soil Compaction and Its Impact on Earthwork Acceptance Testing. *Transportation Research Record: Journal of the Transportation Research Board.*

**Preprints:**

- **Gao, Q.**, Wang, X. (2021). On the Latent Space of Generative Models. *Preprint.*

- Chen, Y., **Gao, Q.**, Wang, X. (2021). iWGAN: an Encoder-Decoder WGAN for Inference. *Under revision at Journal of the Royal Statistical Society: Series B.*

- Mo, Z., Chen, H., **Gao, Q.**, Wang, X. (2020). Uniform Generalization Bound for Generative Adversarial Networks. *Preprint.*

- **Gao, Q.**, Cai, J., Cai, H., Chun, H., Nantung, T. (2019). Risk Control in Acceptance Testing with Percent Within Limit. *Preprint.*