Journal of the American Statistical Association

ISSN: 0162-1459 (Print) 1537-274X (Online) Journal homepage: www.tandfonline.com/journals/uasa20

Adaptive Learning of the Latent Space of Wasserstein Generative Adversarial Networks

Yixuan Qiu, Qingyi Gao & Xiao Wang

To cite this article: Yixuan Qiu, Qingyi Gao & Xiao Wang (2025) Adaptive Learning of the Latent Space of Wasserstein Generative Adversarial Networks, Journal of the American Statistical Association, 120:550, 1254-1266, DOI: 10.1080/01621459.2024.2408778

To link to this article: https://doi.org/10.1080/01621459.2024.2408778

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2025, VOL. 120, NO. 550, 1254–1266: Theory and Methods
https://doi.org/10.1080/01621459.2024.2408778

Adaptive Learning of the Latent Space of Wasserstein Generative Adversarial Networks

Yixuan Qiu[a,*], Qingyi Gao[b,*], and Xiao Wang[b]

[a]School of Statistics and Management, Shanghai University of Finance and Economics, Shanghai, P.R. China; [b]Department of Statistics, Purdue University, West Lafayette, IN

ABSTRACT

Generative models based on latent variables, such as generative adversarial networks (GANs) and variational auto-encoders (VAEs), have gained lots of interests due to their impressive performance in many fields. However, many data such as natural images usually do not populate the ambient Euclidean space but instead reside in a lower-dimensional manifold. Thus an inappropriate choice of the latent dimension fails to uncover the structure of the data, possibly resulting in mismatch of latent representations and poor generative qualities. Toward addressing these problems, we propose a novel framework called the latent Wasserstein GAN (LWGAN) that fuses the Wasserstein auto-encoder and the Wasserstein GAN so that the intrinsic dimension of the data manifold can be adaptively learned by a modified informative latent distribution. We prove that there exist an encoder network and a generator network in such a way that the intrinsic dimension of the learned encoding distribution is equal to the dimension of the data manifold. We theoretically establish that our estimated intrinsic dimension is a consistent estimate of the true dimension of the data manifold. Meanwhile, we provide an upper bound on the generalization error of LWGAN, implying that we force the synthetic data distribution to be similar to the real data distribution from a population perspective. 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 sampling from the learned latent distribution. Supplementary materials for this article are available online, including a standardized description of the materials available for reproducing the work.

ARTICLE HISTORY
Received March 2022
Accepted September 2024

KEYWORDS
Consistency; Generalization error; Generative adversarial networks; Latent variable models; Manifold learning; Minimax optimization; Wasserstein distance

1. Introduction

Unsupervised generative models receive great attentions in the machine learning community nowadays due to their impressive performance in many fields (Kingma and Welling 2014; Goodfellow et al. 2014; Li, Swersky, and Zemel 2015; Dinh, Sohl-Dickstein, and Bengio 2016; Gao et al. 2020; Qiu and Wang 2021). Given a random sample from a p-dimensional random vector $X \in \mathcal{X} \subset \mathbb{R}^p$ with an unknown distribution P_X, the goal is to train a generative model that can produce synthetic data that look similar to the observed samples from X. While there are several ways of quantifying the similarity, the most common approach is to directly employ some of the known divergence measures, such as the Kullback–Leibler (KL) divergence and the Wasserstein distance, between the real data distribution and the synthetic data distribution.

There are two influential frameworks for generative models: generative adversarial networks (GANs, Goodfellow et al. 2014) and variational auto-encoders (VAEs, Kingma and Welling 2014). They are 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 the standard multivariate normal distribution $P_Z = N(0, I_d)$. Then the synthetic data are generated by either a deterministic transformation $G : \mathcal{Z} \to \mathcal{X}$ or a conditional distribution $p(x|z)$ of X given Z.

GAN and WGAN. Training GANs is like a two-player game, where two networks, a generator and a discriminator, are simultaneously trained to allow the powerful discriminator to distinguish between real data and generated samples. As a result, the generator is trying to maximize its probability of having its outputs recognized as real. This leads to the following minimax objective function,

$$\inf_{G\in\mathcal{G}} \sup_{f\in\mathcal{F}} \mathbb{E}_X\left[\log(f(X))\right] + \mathbb{E}_Z\left[\log\left(1 - f(G(Z))\right)\right], \quad (1)$$

where $f \in \mathcal{F}$ is a discriminator and $G \in \mathcal{G}$ is a generator. Optimizing (1) is equivalent to minimizing the Jensen–Shannon divergence between the generation distribution and real data distribution. GANs can generate visually realistic images, but suffer from unstable training and mode collapsing.

The Wasserstein GAN (WGAN, Arjovsky, Chintala, and Bottou 2017) is an extension to the vanilla GAN that improves the stability of training by leveraging the 1-Wasserstein distance between two probability measures. Denote by $P_{G(Z)}$ the genera-

CONTACT Xiao Wang wangxiao@purdue.edu Department of Statistics, Purdue University, West Lafayette, IN 47907.
*These authors contributed equally to this work.
Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JASA.

tion distribution measure, and then the 1-Wasserstein distance between P_X and $P_{G(Z)}$ is defined as

$$W_1(P_X, P_{G(Z)}) = \inf_{\pi \in \Pi(P_X, P_Z)} \mathbb{E}_{(X,Z)\sim\pi} \|X - G(Z)\|, \quad (2)$$

where $\|\cdot\|$ represents the ℓ_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, respectively. It is hard to find the optimal coupling π through this constrained primal problem. However, thanks to the Kantorovich–Rubinstein duality (Villani 2008), WGAN can learn the generator G by minimizing a dual form of (2),

$$W_1(P_X, P_{G(Z)}) = \sup_{f\in\mathcal{F}} \left\{\mathbb{E}_X f(X) - \mathbb{E}_Z f(G(Z))\right\}, \quad (3)$$

where f is called the critic function, and $\mathcal{F}$ is the set of all bounded 1-Lipschitz functions. Weight clipping (Arjovsky, Chintala, and Bottou 2017) and gradient penalty (Gulrajani et al. 2017) are two common strategies to maintain the Lipschitz continuity of f. Weight clipping uses 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 the parameter c. Instead, gradient penalty adds a regularization term, $\mathbb{E}_{\hat{X}}\left\{(\|\nabla_x f(\hat{X})\| - 1)^2\right\}$, to the loss function to enforce the 1-Lipschitz condition, where $\hat{X}$ is sampled uniformly along the segment between pairs of points sampled from P_X and $P_{G(Z)}$. This is motivated by the fact that the optimal f has unit gradient norm on the segment between optimally coupled points from P_X and $P_{G(Z)}$.

VAE and WAE. A VAE defines a "probabilistic decoder" $p_\theta(x|z)$ with the unknown parameter θ. Then the marginal distribution of X is $p_\theta(x) = \int p_\theta(x|z)p_Z(z)\mathrm{d}z$, where $p_Z(\cdot)$ is the density of P_Z. Due to the intractability of this integration, the maximum likelihood estimation is prohibited. Instead, a "probabilistic encoder" $q_\phi(z|x)$ with the unknown parameter ϕ is defined to approximate the posterior distribution $p_\theta(z|x) = p_\theta(x|z)p_Z(z)/p_\theta(x)$. The objective of VAE is to maximize a lower bound of the log-likelihood $\log p_\theta(x)$, which is called the evidence lower bound (ELBO):

$$\mathrm{ELBO} = \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - \mathrm{KL}\left(q_\phi(z|x)\|p_Z(z)\right),$$

where the first term can be efficiently estimated by the Monte Carlo sampling, and the second term has a closed-form expression when q_ϕ 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 posterior.

The Wasserstein auto-encoder (WAE, Tolstikhin et al. 2018) makes two modifications to VAE. It uses a deterministic encoder $Q : \mathcal{X} \to \mathcal{Z}$ to approximate the conditional distribution of Z given X, and a deterministic generator $G : \mathcal{Z} \to \mathcal{X}$ to approximate the conditional distribution of X given Z. In addition, WAE adopts the 1-Wasserstein distance between the real data distribution P_X and the generation distribution $P_{G(Z)}$, rather than the KL divergence used in VAEs, to train the model. Let $P_{Q(X)}$ denote the aggregated posterior distribution measure, and then WAE minimizes the following reconstruction error with respect to the generator G,

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

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 forces the aggregated posterior $P_{Q(X)}$ to match the prior distribution P_Z.

There are several limitations for the generative models above. 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. Another type of generative model called normalizing flows (Dinh, Sohl-Dickstein, and Bengio 2016) keeps the latent dimension the same as the dimension of the data. This is because normalizing flows approximate the data distribution by a deterministic *invertible* mapping G such that $X = G(Z)$. Since many observed data such as natural images lie on a low-dimensional manifold embedded in a higher-dimensional Euclidean space, an inappropriate choice of the latent dimension could cause a wrong latent representation that does not populate the full ambient space (Rubenstein, Schoelkopf, and Tolstikhin 2018). Hence, the wrongly specified latent dimension fails to uncover the structure of the data, and the corresponding generative models may suffer from mode collapsing, under-fitting, mismatch of representation learning, and poor generation qualities. Furthermore, although there are many interesting works taking advantages of both VAEs and GANs (Larsen et al. 2016; Dumoulin et al. 2017; Donahue, Krähenbühl, and Darrell 2017; Chen, Gao, and Wang 2021), it remains unclear what principles are underlying the framework combining the best of WAEs and WGANs when the latent dimension is unknown.

To handle the aforementioned drawbacks, we propose a novel framework, called the latent Wasserstein GAN (LWGAN), to identify the intrinsic dimension of a data distribution that lies on a topological manifold, and then improve the quality of generative modeling as well as representation learning. We have performed two major modifications to the current GAN and VAE frameworks. First, we change the latent distribution from $N(0, I_d)$ to a generalized normal distribution $N(0, A)$ with A being a diagonal matrix with entries taking values 0 or 1. Therefore, the rank of A allows us to characterize the intrinsic dimension of the latent space. This modification has been adopted for the flow model to reduce the dimension of the latent space (Zhang et al. 2023), but it has not been applied to GAN or VAE models. Second, we combine WGAN and WAE in a principled way motivated by the primal-dual iterative algorithm. We utilize a deterministic encoder $Q : \mathcal{X} \to \mathcal{Z}$ to learn an informative prior distribution $P_Z \sim N(0, A)$. On the other hand, a generator $G : \mathcal{Z} \to \mathcal{X}$ is combined with Q to generate images that look like the real ones using the latent code Z from P_Z. We theoretically guarantee the existence of such a generator G and an encoder Q. To get rid of possible invalid divergences, we focus on the 1-Wasserstein distance to measure the similarities between two distributions, which applies to any pair of distributions as long as they can be sufficiently sampled. Note that the KL divergence is not well-defined when the supports of two probability measures do not overlap, which is very common for high-dimensional data.

The rest of the article is organized as follows. Section 2 investigates the phenomenon of dimension mismatch between the latent distribution and data distribution. Section 3 presents the

new LWGAN framework that provides a feasible way to estimate the encoder, generator, and intrinsic dimension. Theoretical analyses are given in Section 4, including results on generalization error bounds, estimation consistency, and intrinsic dimension consistency. Section 5 demonstrates extensive numerical experiments under different settings to verify that the LWGAN is able to detect the intrinsic dimensions for both simulated examples and real image data. Finally, Section 6 concludes this article. Proofs of theorems and additional numerical results are provided in the supplementary materials.

2. Issues of Latent Dimension Mismatch

Throughout this article we use $\mathcal{X} \subset \mathbb{R}^p$ and $\mathcal{Z} \subset \mathbb{R}^d$ to denote the spaces of observed data points and latent variables, respectively. To precisely describe the structure of high-dimensional data with a low latent dimension, we first make the following definition of a topological manifold.

Definition 1 (Topological manifold, Lee 2013). Suppose that $\mathcal{M}$ is a topological space. $\mathcal{M}$ is a topological manifold of dimension r if $\mathcal{M}$ is a second-countable Hausdorff space, and for each $x \in \mathcal{M}$, there exist an open subset $U \subset \mathcal{M}$ containing x, an open subset $V \subset \mathbb{R}^r$, and a homeomorphism φ between U and V. A homeomorphism $\varphi : U \to V$ is a continuous bijective mapping with a continuous inverse φ^{-1}.

In this article, all manifolds are referred to as topological manifolds unless otherwise noted. Typically, $\mathcal{M}$ is a subset of some Euclidean space $\mathbb{R}^p$, in which case the Hausdorff and second-countability properties in Definition 1 are automatically inherited from the Euclidean topology. To exclude overly complicated cases, we moderately strengthen the qualification of the homeomorphism φ in Definition 1 to make it a global one:

Assumption 1. $\mathcal{X}$ is an r-dimensional manifold, and there exists a homeomorphism φ between $\mathcal{X}$ and $\mathbb{R}^r$.

In what follows, the symbol φ is used to denote one homeomorphism between $\mathcal{X}$ and $\mathbb{R}^r$. Then we can define a continuous distribution supported on the manifold $\mathcal{X}$ that satisfies Assumption 1.

Definition 2. A random vector $X \in \mathbb{R}^p$ is said to have a continuous distribution P_X supported on $\mathcal{X}$, if its image $\varphi(X)$ follows a continuous distribution on $\mathbb{R}^r$.

Let $X \in \mathcal{X} \subset \mathbb{R}^p$ be the observed data with a continuous distribution P_X supported on $\mathcal{X}$, where $\mathcal{X}$ satisfies Assumption 1. We define the intrinsic dimension of the data distribution P_X as the dimension of the manifold $\mathcal{X}$, denoted by $\texttt{InDim}(P_X) = r$, and its ambient dimension as the dimension of the enclosing Euclidean space, denoted by $\texttt{AmDim}(P_X) = p$. By Theorem 1.2 of Lee (2013), $\texttt{InDim}(P_X)$ must be unique, and it cannot be larger than $\texttt{AmDim}(P_X)$.

In most existing deep generative models, the latent variable Z is selected as a d-dimensional standard normal distribution $N(0, I_d)$, so $\texttt{InDim}(P_Z) = \texttt{AmDim}(P_Z) = d$. The dimension d is typically predetermined to be a number that is smaller than p. In GAN-based models, if the generator G is a continuous function, then the synthetic sample $G(Z)$ will be supported on a manifold of dimension at most $\texttt{InDim}(P_Z)$. When $\texttt{InDim}(P_Z) < \texttt{InDim}(P_X)$, forcing $P_{G(Z)}$ to be close to P_X with unmatched intrinsic dimensions is a challenging task. On the other hand, in auto-encoder-based models, similar phenomenon of dimension mismatch occurs for the encoded distribution $P_{Q(X)}$. For example, it is difficult to enforce $P_{Q(X)}$ to be close to P_Z if $\texttt{InDim}(P_X) < \texttt{InDim}(P_Z)$, as filling a plane with a one-dimensional curve is hard.

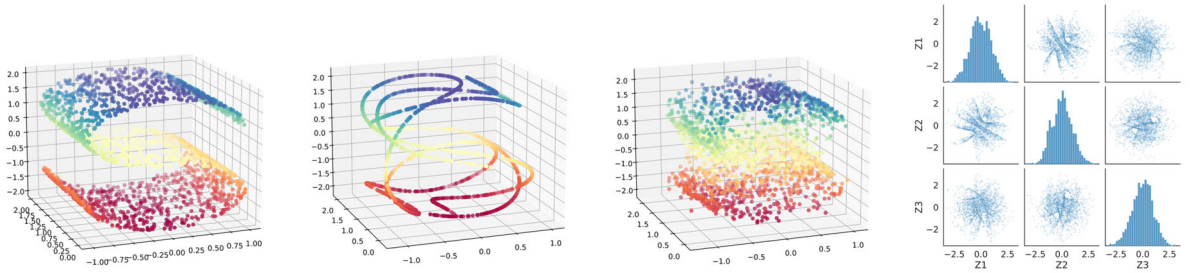
To highlight this phenomenon and to motivate our proposed model, we first employ a toy example to provide intuitions for the effects and consequences of different intrinsic dimensions of the model and data distributions. Consider a 3D S-curve dataset as shown in Figure 1(a), where each data point $X = (X_1, X_2, X_3)$ is generated by

$$X_1 = \sin(3\pi(U - 0.5)), \quad X_2 = 2V,$$
$$X_3 = \text{sign}(3\pi(U - 0.5))\cos(3\pi(U - 0.5)),$$

for $U \sim Unif(0, 1)$ and $V \sim N(0, 1)$. This example results in $\texttt{AmDim}(P_X) = 3$ and $\texttt{InDim}(P_X) = 2$. We first choose the latent distribution P_Z to be a one-dimensional normal distribution $N(0, 1)$, and then the generated sample from WGAN is plotted in Figure 1(b). To minimize the 1-Wasserstein distance between the real distribution P_X and the generation distribution $P_{G(Z)}$, WGAN learns an outer contour of the S-curve, but it cannot fill points on the surface. Instead, if we choose a three-dimensional standard normal $N(0, I_3)$ as the latent distribution, then WAE is forced to reconstruct the images well, but at the same time it tries to fill the three-dimensional latent space evenly by a distribution supported on a two-dimensional manifold. The only way to do this is by curling the manifold up in the latent space as shown in Figure 1(d). This disparity between P_Z and $P_{Q(X)}$ in the latent space induces a poor generation of $P_{G(Z)}$ in Figure 1(c).

3. The Latent Wasserstein GAN

A natural solution to the mismatch problem described in Section 2 is to select a latent distribution P_Z whose intrinsic dimension is the same as that of the data distribution P_X. However, $\texttt{InDim}(P_X)$ is typically unknown, so one option is to learn it from the data. When both the continuous generator G and the continuous encoder Q are combined in an auto-encoder generative model, $P_{G(Z)} = P_X$ and $P_{Q(X)} = P_Z$ cannot be satisfied simultaneously unless $\texttt{InDim}(P_X) = \texttt{InDim}(P_Z)$ according to our previous discussion. This motivates us to search for an encoder Q and a corresponding generator G, such that $Q(X)$ reflects the latent space supported on an r-dimensional manifold, and generated samples using the latent variables are of high quality. To be concrete, we need an auto-encoder generative model that satisfies the following three goals at the same time: (a) the latent distribution P_Z is supported on an r-dimensional manifold; (b) the distribution of $G(Z)$ is similar to P_X; (c) the difference between X and its reconstruction $G(Q(X))$ is small.



(a) S-curve data (b) WGAN: Generation (c) WAE: Generation (d) WAE: Latent space

Figure 1. Illustrations of data generation with wrong latent dimensions in WGAN and WAE.

3.1. Existence of Optimal Encoder-Generator Pairs

Unlike conventional generative models that use a fixed standard normal distribution as the latent distribution, we consider a latent distribution whose intrinsic dimension could be less than d, that is, the latent variable $Z \in \mathcal{Z} \subset \mathbb{R}^d$ can have a distribution supported on some manifold $\mathcal{Z}$. This idea is realized by the generalized definition of the normal distribution (Zhang et al. 2023). In particular, let $A_s = \text{diag}(1, \ldots, 1, 0, \ldots, 0)$ be a diagonal matrix whose first s diagonal elements are one and whose remaining $(d - s)$ diagonal elements are zero, and Z_0 be a random vector following standard multivariate normal distribution $N(0, I_d)$. Then clearly, the random vector $Z = A_s Z_0$ is supported on an s-dimensional manifold $\mathcal{Z}$, and its distribution $P_Z \equiv P_{A_s Z_0}$ has dimensions $\texttt{InDim}(P_Z) = s$ and $\texttt{AmDim}(P_Z) = d$. For convenience, we use the classic notation $N(0, A_s)$ to denote this distribution, although A_s is a degenerate covariance matrix.

Choosing $P_Z = N(0, A_s)$, where s is a parameter to estimate, enables us to solve the dimension mismatch problem in Section 2. If $s = r$, then the latent variable Z can be mapped to $G(Z)$ supported on an r-dimensional manifold, and meanwhile, P_Z and the encoded distribution $P_{Q(X)}$ can have matched intrinsic dimensions. Formally, Theorem 1 states that for any data distribution P_X defined by Definition 2, there always exist a *continuous* encoder $Q^\diamond$ that guarantees meaningful encodings on an r-dimensional manifold, and a *continuous* generator $G^\diamond$ that generates samples with the same distribution as P_X, using those latent points encoded by $Q^\diamond$.

Theorem 1. If $d \geq r$, then there exist two continuous mappings $Q^\diamond : \mathcal{X} \to \mathcal{Z}$ and $G^\diamond : \mathcal{Z} \to \mathcal{X}$ such that $Q^\diamond(X) \sim N(0, A_r)$ and $X = G^\diamond(Q^\diamond(X))$.

In such cases, we call $(Q^\diamond, G^\diamond)$ an *optimal* encoder-generator pair for the data distribution P_X, and note that $(Q^\diamond, G^\diamond)$ may not be unique. On the other hand, Corollary 1 shows that if the ambient dimension of P_Z is insufficient, then the auto-encoder structure is unable to recover the original distribution of X, which justifies the finding in Figure 1(b).

Corollary 1. Suppose that $d < r$. Then for any continuous mappings $Q : \mathbb{R}^p \to \mathbb{R}^d$ and $G : \mathbb{R}^d \to \mathbb{R}^p$, we have $\mathbb{E}_X \|X - G(Q(X))\| > 0$.

3.2. The Proposed Model

Theorem 1 shows the possibility to identify the dimension of the data manifold $\mathcal{X}$ by learning a latent distribution with the same intrinsic dimension via the encoder Q. In this section, we realize this idea through our new auto-encoder generative model, LWGAN, which takes advantages of both WGAN and WAE. LWGAN is capable of learning Q, G, and r simultaneously to accomplish all of our three goals. For brevity, we abbreviate the subscript s in the matrix A_s when no confusion is caused.

There are three probability measures involved in our problem: the real data distribution P_X, the generation distribution $P_{G(AZ_0)}$, and the reconstruction distribution $P_{G(Q(X))}$. Our goal is to ensure that all three measures are similar to each other in a systematic way. To this end, we propose the following distance between P_X and $P_{G(AZ_0)}$ with given G and A:

$$\overline{W}_1(P_X, P_{G(AZ_0)}) = \inf_{Q \in \mathcal{Q}^\diamond} \sup_{f \in \mathcal{F}^\diamond} \mathfrak{L}_A(G, Q, f), \tag{4}$$

$$\mathfrak{L}_A(G, Q, f) = \mathbb{E}_X \|X - G(Q(X))\| + \mathbb{E}_X \left[f(G(Q(X)))\right] - \mathbb{E}_{Z_0} \left[f(G(AZ_0))\right],$$

where $\mathcal{F}^\diamond$ is the set of all bounded 1-Lipschitz functions, and $\mathcal{Q}^\diamond$ is the set of continuous encoder mappings. The term $\mathbb{E}_X \|X - G(Q(X))\|$ can be viewed as the auto-encoder reconstruction error in WAE, and also a loss to measure the discrepancy between P_X and $P_{G(Q(X))}$. The other term $\mathbb{E}_X \left[f(G(Q(X)))\right] - \mathbb{E}_{Z_0} \left[f(G(AZ_0))\right]$ quantities the difference between $P_{G(Q(X))}$ and $P_{G(AZ_0)}$. Theorem 2 shows that, under some mild conditions, (4) achieves its minimum as the 1-Wasserstein distance $W_1(P_X, P_{G(AZ_0)})$.

Theorem 2. The $\overline{W}_1$ distance defined in (4) has the following representation:

$$\overline{W}_1(P_X, P_{G(AZ_0)}) = \inf_{Q \in \mathcal{Q}^\diamond} \left\{ W_1(P_X, P_{G(Q(X))}) + W_1(P_{G(Q(X))}, P_{G(AZ_0)}) \right\}. \tag{5}$$

Therefore, $W_1(P_X, P_{G(AZ_0)}) \leq \overline{W}_1(P_X, P_{G(AZ_0)})$, and the equality holds if there exists an encoder $Q \in \mathcal{Q}^\diamond$ such that $Q(X)$ has the same distribution as AZ_0.

Remark 1. Theorem 1 shows that there exists some optimal encoder-generator pair $(Q^\diamond, G^\diamond)$ such that $Q^\diamond(X) \overset{d}{=} A_r Z_0$ and $X = G^\diamond(Q^\diamond(X))$. Therefore, $Q^\diamond$ is an optimal solution to (5) for $A = A_r$, and hence the equality $W_1(P_X, P_{G(A_r Z_0)}) = \overline{W}_1(P_X, P_{G(A_r Z_0)})$ holds. This indicates that $\overline{W}_1$ is a tight upper bound for W_1. Furthermore, with $G = G^\diamond$, we have $\overline{W}_1(P_X, P_{G^\diamond(A_r Z_0)}) = 0$, which reaches its global minimum.

Remark 2. The condition $Q(X) \overset{d}{=} AZ_0$ is sufficient but not necessary for $W_1 = \overline{W}_1$ to hold. For example, using $(Q^\diamond, G^\diamond)$ in the proof of Theorem 1, we can show that $Q^\diamond(X) \overset{d}{=} A_r Z_0$ but $W_1(P_X, P_{G^\diamond(A_s Z_0)}) = \overline{W}_1(P_X, P_{G^\diamond(A_s Z_0)}) = 0$ for any s such that $r \le s \le d$.

In our framework, we represent the encoder, generator, and critic using deep neural networks, $G = G(\cdot;\theta_G)$, $Q = Q(\cdot;\theta_Q)$, $f = f(\cdot;\theta_f)$, where $\theta = (\theta_G, \theta_Q, \theta_f)$ are the network parameters. We restrict the three components of θ to compact sets Θ_G, Θ_Q, and Θ_f, respectively, and further define $\bar{\Theta}_f = \{\theta_f \in \Theta_f : \|f(\cdot;\theta_f)\|_L \le 1\}$, where $\|g\|_L$ stands for the Lipschitz constant of a function g. Then we define the parameter space $\Theta = \Theta_G \times \Theta_Q \times \bar{\Theta}_f$ and function spaces $\mathcal{G} = \{G(\cdot;\theta_G) : \theta_G \in \Theta_G\}$, $\mathcal{Q} = \{Q(\cdot;\theta_Q) : \theta_Q \in \Theta_Q\}$, $\mathcal{F} = \{f(\cdot;\theta_f) : \theta_f \in \bar{\Theta}_f\}$. Accordingly, hereafter we replace the spaces $\mathcal{Q}^\diamond$ and $\mathcal{F}^\diamond$ in (4) with $\mathcal{Q}$ and $\mathcal{F}$, respectively for the definition of $\overline{W}_1(P_X, P_{G(AZ_0)})$.

In practice, we only have the empirical versions of P_X and $P_{G(AZ_0)}$. Suppose we have observed an iid data sample $X_1, \ldots, X_n$, and have simulated an iid sample of $N(0, I_d)$, $Z_{0,1}, \ldots, Z_{0,n}$, where X and Z_0 samples are independent. Then we define

$$\begin{aligned} L(x,z;\theta) &= \|x - G(Q(x;\theta_Q);\theta_G)\| \\ &\quad + f(G(Q(x;\theta_Q);\theta_G);\theta_f) - f(G(z;\theta_G);\theta_f), \\ \ell(\theta, A) &= \mathbb{E}_{X\otimes Z_0}[L(X, AZ_0, \theta)], \\ \hat{\ell}_n(\theta, A) &= \frac{1}{n}\sum_{i=1}^n L(X_i, AZ_{0,i}, \theta), \end{aligned}$$

where $\mathbb{E}_{X\otimes Z_0}$ means taking the expectation of independent X and Z_0. Clearly,

$$\begin{aligned} \overline{W}_1(P_X, P_{G(AZ_0)}) &= \inf_{Q\in\mathcal{Q}} \sup_{f\in\mathcal{F}} \mathfrak{L}(G, Q, f, A) \\ &= \inf_{\theta_Q\in\Theta_Q} \sup_{\theta_f\in\bar{\Theta}_f} \ell(\theta, A), \end{aligned}$$

and we denote its empirical version as $\overline{W}_1(\hat{P}_X, \hat{P}_{G(AZ_0)}) = \inf_{\theta_Q\in\Theta_Q} \sup_{\theta_f\in\bar{\Theta}_f} \hat{\ell}_n(\theta, A)$.

Remark 1 of Theorem 2 motivates us to estimate the generator G and the rank-revealing matrix A based on the $\overline{W}_1$ distance, but Remark 2 suggests that purely minimizing $\overline{W}_1$ is not enough, since a matrix A with a rank larger than r can still drive $\overline{W}_1$ to zero, the global minimum value. Therefore, we also need to introduce a penalty term to regularize the rank of A. Since A is uniquely determined by its rank s, below A and s are used interchangeably to represent the rank parameter. Define the rank-regularized objective function as

$$\hat{\rho}_n(\theta_G, A) = \overline{W}_1(\hat{P}_X, \hat{P}_{G(AZ_0)}) + \lambda_n \cdot \mathrm{rank}(A),$$

where λ_n is a deterministic sequence satisfying $\lambda_n \to 0$ and $n^{1/2}\lambda_n \to \infty$, which will be justified in Theorem 5. Then the generator G and the matrix A are estimated by

$$(\hat{\theta}_G, \hat{r}) = \operatorname*{arg\,min}_{\theta_G\in\Theta_G, 1\le s\le d} \hat{\rho}_n(\theta_G, A_s). \tag{6}$$

When the optimal points are not unique, $\hat{\theta}_G$ can be chosen arbitrarily from the solution set, and $\hat{r}$ is taken as the smallest one among all the optimal points.

3.3. Computational Algorithm

The optimization problem (6) can be solved by computing the "rank score"

$$\hat{\varrho}_n(s) = \min_{\theta_G,\theta_Q} \max_{\theta_f} \hat{\ell}_n(\theta, A_s) + \lambda_n s \tag{7}$$

for each $s = 1, \ldots, d$, and then we have $\hat{r} = \operatorname{arg\,min}_s \hat{\varrho}_n(s)$. Equivalently, we need to solve

$$\begin{aligned} \min_{G_1,Q_1} \max_{f_1} \ & \frac{1}{n}\sum_{i=1}^n \big[\|X_i - G_1(Q_1(X_i))\| + f_1(G_1(Q_1(X_i))) \\ & - f_1(G_1(A_1 Z_{0,i}))\big] + \lambda_n \cdot 1 \\ \cdots \quad & \cdots \\ \min_{G_d,Q_d} \max_{f_d} \ & \frac{1}{n}\sum_{i=1}^n \big[\|X_i - G_d(Q_d(X_i))\| + f_d(G_d(Q_d(X_i))) \\ & - f_d(G_d(A_d Z_{0,i}))\big] + \lambda_n \cdot d \end{aligned} \tag{8}$$

by fitting d different sets of neural networks (G_s, Q_s, f_s), $s = 1, \ldots, d$, which may be time-consuming. Instead, we propose a practical and efficient algorithm based on the idea that encoder and critic functions under different ranks can share network parameters. We slightly modify the network structures of $Q(x;\theta_Q)$ and $f(x;\theta_f)$ such that they also receive a rank input e_s, where the one-hot encoding vector e_s is the sth column of the identity matrix I_d. As a result, the rank-aware encoder and critic functions become $Q(x, e_s;\theta_Q)$ and $f(x, e_s;\theta_f)$, respectively. We also make the output of $Q(x, e_s;\theta_Q)$ to have rank s by setting the last $(d - s)$ components to zero. The generator G does not need this modification, since its input $Q(X, e_s)$ or $A_s Z_0$ already contains the rank information.

Then problem (9) is equivalent to solving

$$\begin{aligned} \min_{G,Q} \max_{f} \ & \frac{1}{nd}\sum_{s=1}^d\sum_{i=1}^n [\|X_i - G(Q(X_i, e_s))\| \\ & + f(G(Q(X_i, e_s)), e_s) - f(G(A_s Z_{0,i}), e_s)\big], \end{aligned} \tag{9}$$

as long as the rank-aware neural networks (G, Q, f) have sufficient expressive powers. This would be a reasonable assumption if we recognize that (G_s, Q_s, f_s) and (G_t, Q_t, f_t) should be similar if $s \approx t$. In practice, this means that (G_s, Q_s, f_s) and (G_t, Q_t, f_t) can share most of the neural network parameters, and their difference is reflected by the input rank information e_s. Also note that the rank penalty terms in (9) are tentatively dropped, since they only affect the estimation of s but not (G, Q, f). The rank terms will be added back once the optimal (G, Q, f) are obtained.

Furthermore, the objective function of (9) can be viewed as an empirical expectation over (X, Z, S), where the average term $d^{-1}\sum_{s=1}^d(\cdot)$ represents an expectation $\mathbb{E}_S(\cdot)$ with S following a discrete uniform distribution on $\{1, \ldots, d\}$. Therefore, to further save computing time, we can randomly pick a rank in each iteration, and then update (G, Q, f) accordingly. In our numerical experiments, we have saved various metrics to monitor the training processs, and they demonstrate that this computing algorithm is both stable and efficient (see Section S2.3 of the supplementary material).

The training details are summarized in Algorithm 1. In our algorithm, the 1-Lipschitz constraint on the critic f is

Algorithm 1 The training algorithm of LWGAN.

Input: Initial parameter value $\theta^{(0)}$, batch size M, critic update frequency L, gradient penalty parameter λ_{GP}, rank regularization parameter λ_n.

Output: Neural network parameters $\hat{\theta}$, estimated intrinsic dimension $\hat{r}$.

for $k = 1, 2, \ldots, T$ **do**
  Randomly select an integer s from $1, \ldots, d$ with equal probabilities
  Set $\theta^{(k,0)} \leftarrow \theta^{(k-1)}$
  for $l = 1, 2, \ldots, L$ **do**
    Sample real data $X_1, \ldots, X_M \overset{\text{iid}}{\sim} P_X$, latent data $Z_{0,1}, \ldots, Z_{0,M} \overset{\text{iid}}{\sim} N(0, I_d)$, and $\varepsilon_1, \ldots, \varepsilon_M \overset{\text{iid}}{\sim} \text{Unif}(0,1)$
    Set $\hat{X}_i = \varepsilon_i X_i + (1-\varepsilon_i) G(A_s Z_{0,i}; \theta_G^{(k)})$, $i = 1, \ldots, M$
    Define $J(\theta) = \hat{\ell}_M(\theta, A_s) + \lambda_{\text{GP}} \cdot M^{-1} \sum_{i=1}^{M} \left(\|\nabla_x f(\hat{X}_i; \theta_f)\| - 1 \right)^2$
    Update $\theta_f^{(k,l)} \leftarrow \theta_f^{(k,l-1)} + \text{Adam}\left(-\nabla_{\theta_f} J(\theta)|_{\theta=\theta^{(k,l-1)}} \right)$
  end for
  Sample real data $X_1, \ldots, X_M \overset{\text{iid}}{\sim} P_X$ and latent data $Z_{0,1}, \ldots, Z_{0,M} \overset{\text{iid}}{\sim} N(0, I_d)$
  Update $\theta_{G,Q}^{(k)} \leftarrow \theta_{G,Q}^{(k,L)} + \text{Adam}\left(\nabla_{\theta_{G,Q}} \hat{\ell}_M(\theta, A_s)|_{\theta=\theta^{(k,L)}} \right)$
  if $\theta^{(k)}$ converges **then**
    Compute $\hat{\varrho}_n(s) = \hat{\ell}_n(\theta^{(k)}, A_s) + \lambda_n s$, $s = 1, \ldots, d$
    return $\hat{\theta} = \theta^{(k)}$, $\hat{r} = \arg\min_s \hat{\varrho}_n(s)$
  end if
end for

enforced by the gradient penalty technique proposed in Gulrajani et al. (2017), where $\hat{X}$ is sampled uniformly along the segment between pairs of points sampled from P_X and $P_{G(AZ_0)}$, and λ_{GP} is the regularization level of the gradient penalty. The operator Adam$(\cdot)$ means applying the Adam optimization method (Kingma and Ba 2014) to update neural network parameters θ.

3.4. Tuning Parameter Selection

Another critical issue in applying LWGAN to real-life data is the selection of the regularzation parameter λ_n in (7). From a theoretical perspective, in Section 4 we will show that λ_n should be chosen such that $\lambda_n \to 0$ and $n^{1/2}\lambda_n \to \infty$, whereas in this section, we propose a more practical and data-driven scheme for selecting λ_n. The intuition is to note that without the rank penalty, $\hat{V}_n(A_s) \coloneqq \hat{\varrho}_n(s) - \lambda_n s$ would all be close to zero for $s \geq r$, and their differences are mainly attributed to the randomness from estimation. Therefore, if we can estimate the standard errors of $\hat{V}_n(A_s)$ for $s \geq r$, then λ_n should be chosen slightly larger than the estimated standard error, so as to encourage the selection of the simplest model, namely, the model with the smallest rank s.

Concretely, we use the following method to determine the data-driven λ_n. First, train the model to optimum according to Algorithm 1, using the whole training dataset. Second, continue to train the model for $\tilde{T}$ iterations, using a subset of the training data, denoted as $\tilde{X}_1$. This can be viewed as fitting a model on $\tilde{X}_1$ based on a warm start. Third, based on this model, compute the metric $\hat{V}_n(A_s)$ for each s, and we use the symbol $\hat{V}_{1s}$ to denote its value. Then repeat this process on different training data subsets $\tilde{X}_k$, $k = 2, \ldots, \tilde{K}$, and similarly compute the scores $\hat{V}_{ks}$, $k = 2, \ldots, \tilde{K}$, $s = 1, \ldots, d$. Let

$$\tilde{r} = \arg\min_s \hat{V}_{\cdot s} \coloneqq \frac{1}{\tilde{K}} \sum_{k=1}^{\tilde{K}} \hat{V}_{ks},$$

$$\widehat{\text{SE}} = \sqrt{\frac{1}{\tilde{K}-1} \sum_{k=1}^{\tilde{K}} \left(\hat{V}_{k\tilde{r}} - \hat{V}_{\cdot\tilde{r}} \right)^2}.$$

In other words, we first find the rank s that has the smallest mean value $\hat{V}_{\cdot s}$, and then estimate the standard error of the mean on this rank. Finally, we set $\lambda_n = \widehat{\text{SE}}^{0.8}$. In a typical setting, $\widehat{\text{SE}} = O(n^{-1/2})$, so $\lambda_n = O(n^{-0.4})$ satisfies the theoretical rate. Our numerical experiments use $\tilde{T} = 20$ and $\tilde{K} = 50$, so this method essentially trains the model for additional 1000 iterations, which is relatively small compared to the main training cost for real-life datasets.

4. Theoretical Results

4.1. Generalization Error Bound

Since the LWGAN model highly relies on the $\overline{W}_1$ distance, and the estimators are based on its empirical version, a natural question is how well the empirical quantity $\overline{W}_1(\hat{P}_X, \hat{P}_{G(AZ_0)})$ approximates the population quantity $\overline{W}_1(P_X, P_{G(AZ_0)})$. This problem can be characterized by the generalization error. In the context of supervised learning, the generalization error is defined as the gap between the empirical risk (i.e., the training error) and the expected risk (i.e., the testing error). Similarly, in the framework of LWGAN, we make the following definition derived from Arora et al. (2017).

Definition 3. Given $\hat{P}_X$, an empirical version of the true data distribution with n observations, a generation distribution $P_{G(AZ_0)}$ *generalizes* under the $\overline{W}_1(\cdot,\cdot)$ distance with generalization error ε, if

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

holds with a high probability, where $\hat{P}_{G(AZ_0)}$ is an empirical version of the generation distribution $P_{G(AZ_0)}$ with polynomial number of observations drawn after $P_{G(AZ_0)}$ is fixed.

Since the empirical version is what we have access to in practice, a small generalization error implies that after we minimize the empirical $\overline{W}_1$ distance, we can expect a small distance between the true data distribution and the generation distribution. To present the theorem below, we define the function sets $\mathcal{F} \circ G \circ \mathcal{Q} = \{f \circ G \circ Q : f \in \mathcal{F}, Q \in \mathcal{Q}\}$ and $\mathcal{F} \circ G \circ \mathcal{A} = \{h : h(z) = f(G(A_s z)), f \in \mathcal{F}, 1 \leq s \leq d\}$.

Theorem 3. Assume that $\|x\| \leq B$ for all $x \in \mathcal{X}$, and every function in $\mathcal{Q}$ is L_Q-Lipschitz with respect to the input and L_{θ_Q}-Lipschitz with respect to the parameter. For a fixed L_G-Lipschitz

generator G, let $\hat{\Theta}_Q$ be an $\varepsilon/(8L_G L_{\theta_Q})$-net of the encoder parameter space Θ_Q. Then with a probability at least

$$1 - e^{-d} - 2d|\hat{\Theta}_Q| \exp\left\{-\frac{n\varepsilon^2}{8[(1+2L_G L_Q)B + L_G t_{n,d}]^2}\right\},$$

where $t_{n,d} = \sqrt{3d + 2\log n + 2\sqrt{d^2 + d\log n}}$, the following inequality holds:

$$\max_{1\le s\le d}\left|\overline{W}_1(P_X, P_{G(A_sZ_0)}) - \overline{W}_1(\hat{P}_X, \hat{P}_{G(A_sZ_0)})\right| \le 2\mathfrak{R}_n(\mathcal{F}\circ G\circ \mathcal{Q}) + 2\mathfrak{R}_n(\mathcal{F}\circ G\circ \mathcal{A}) + \varepsilon, \quad (10)$$

where $\mathfrak{R}_n(\mathcal{F}\circ G\circ \mathcal{Q}) = \mathbb{E}_\delta\left\{\sup_{f\in\mathcal{F},Q\in\mathcal{Q}} n^{-1}\sum_{i=1}^n \delta_i f(G(Q(X_i)))\right\}$ and $\mathfrak{R}_n(\mathcal{F}\circ G\circ \mathcal{A}) = \mathbb{E}_\delta\left\{\sup_{f\in\mathcal{F},1\le s\le d} n^{-1}\sum_{i=1}^n \delta_i f(G(A_sZ_{0,i}))\right\}$ are Rademacher complexities of the function sets $\mathcal{F}\circ G\circ\mathcal{Q}$ and $\mathcal{F}\circ G\circ\mathcal{A}$, respectively, $\delta = (\delta_1,\ldots,\delta_n)$ are independent Rademacher variables, that is, $P(\delta_i = 1) = P(\delta_i = -1) = 1/2$, and $\mathbb{E}_\delta$ stands for expectations with respect to δ while fixing X and Z_0.

Theorem 3 describes how the function classes $\mathcal{F}$ and $\mathcal{Q}$ contribute to the generalization error bound in our framework. Given a fixed generator G, there exists a uniform upper bound for any critic $f\in\mathcal{F}$, encoder $Q\in\mathcal{Q}$, and low-rank matrix A with appropriate numbers of observations from P_X and P_{Z_0}. More concretely, if $|\hat{\Theta}_Q|$ is small and the sample size is large, then the generalization error is consequently guaranteed to hold with a high probability. In Gao and Wang (2021), it has been proved that $\log(|\hat{\Theta}_Q|) \le \mathcal{O}(K_Q^2 D_Q \log(D_Q L_Q L_G L_{\theta_Q}/\varepsilon))$, where K_Q and D_Q denote the width and depth of Q, respectively. Additionally, the Lipschitz constants of Q and G are under the control of the spectral normalization of their weights.

The Rademacher complexities in (10) measure the 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, $\mathfrak{R}_n(\mathcal{F}\circ G\circ\mathcal{Q})$ is upper bounded by an order scaling as $\mathcal{O}(L_G L_Q\sqrt{(K_Q^2 D_Q + K_f^2 D_f)/n})$, where K_f and D_f denote the width and depth of f, respectively. Similarly, an upper bound on $\mathfrak{R}_n(\mathcal{F}\circ G\circ\mathcal{A})$ scales as $\mathcal{O}(L_G\sqrt{(d^2 + K_f^2 D_f)/n})$ (Gao and Wang 2021).

Finally, since $\overline{W}_1(P_X, P_{G(AZ_0)})$ is a tight upper bound for the 1-Wasserstein distance between P_X and $P_{G(AZ_0)}$ from Theorem 2, we further have

$$W_1(P_X, P_{G(A_sZ_0)}) \le \overline{W}_1(\hat{P}_X, \hat{P}_{G(A_sZ_0)}) + 2\mathfrak{R}_n(\mathcal{F}\circ G\circ\mathcal{Q}) + 2\mathfrak{R}_n(\mathcal{F}\circ G\circ\mathcal{A}) + \varepsilon$$

with a high probability. This implies that from the population perspective, the real data distribution is close to the generation distribution with respective to the 1-Wasserstein distance when we minimize the empirical loss function $\overline{W}_1(\hat{P}_X, \hat{P}_{G(A_sZ_0)})$.

4.2. Estimation Consistency

Theorem 1 has shown that an optimal encoder-generator pair globally minimizes the $\overline{W}_1(P_X, P_{G(AZ_0)})$ distance under a suitable rank of A, and equation (6) indicates that the encoder and generator are estimated by minimizing the empirical version $\overline{W}_1(\hat{P}_X, \hat{P}_{G(AZ_0)})$. Therefore, the question of interest here is how the estimated quantities relate to the population ones.

However, unlike regular parameter estimation problems, an important property of the encoder-generator structure in LWGAN is that the encoder-generator pair may not be unique even with the same objective function value. For example, when Q and G simultaneously permute the first s output and input variables, respectively, the corresponding value of $\mathfrak{L}_A(G,Q,f)$ does not change. Therefore, the optimal solutions to (6) are not singletons but set-valued. In this section, we first fix the rank of A, and consider the estimation consistency through a distance between sets called Hausdorff distance (Rockafellar and Wets 2009). We defer the estimation of the optimal rank of A, or equivalently, $\texttt{InDim}(P_X)$, to Section 4.3.

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

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

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

Recall that we represent G, Q, and f using deep neural networks, and we pre-specify the network structures for these mappings, such as the widths and depths. In this section we only consider functions within the space $\mathcal{G}\times\mathcal{Q}\times\mathcal{F}$. Introduce the function $\phi_A(\theta_G,\theta_Q) = \sup_{\theta_f}\ell(\theta,A)$, and then an optimal solution θ^* solves

$$\inf_{\theta_G}\overline{W}_1(P_X, P_{G(AZ_0)}) = \inf_{\theta_G,\theta_Q}\sup_{\theta_f}\ell(\theta,A) = \inf_{\theta_G,\theta_Q}\phi_A(\theta_G,\theta_Q)$$

when it is a solution to both the outer minimization problem and the inner maximization problem. Therefore, the optimal solution set Θ_A^* is defined as

$$\Theta_A^* = \left\{\theta^*\in\Theta : \phi_A(\theta_G^*,\theta_Q^*) = \inf_{\theta_G,\theta_Q}\phi_A(\theta_G,\theta_Q),\ \ell(\theta^*,A) = \phi_A(\theta_G^*,\theta_Q^*)\right\}.$$

For the empirical minimax problem $\inf_{\theta_G,\theta_Q}\sup_{\theta_f}\hat{\ell}_n(\theta,A)$, algorithms typically search for approximate solutions rather than exact ones. Therefore, we define the empirical solution set with slackness level τ_n as

$$\hat{\Theta}_{n,A}^*(\tau_n) = \left\{\theta^*\in\Theta : \hat{\phi}_A(\theta_G^*,\theta_Q^*) \le \inf_{\theta_G,\theta_Q}\phi_A(\theta_G,\theta_Q) + \tau_n,\ \hat{\ell}_n(\theta^*,A) \ge \hat{\phi}_A(\theta_G^*,\theta_Q^*) - \tau_n\right\},$$

where $\hat{\phi}_A(\theta_G,\theta_Q) = \sup_{\theta_f}\hat{\ell}_n(\theta,A)$, and τ_n is a sequence of nonnegative random variables such that $\tau_n \xrightarrow{P} 0$. We further make some assumptions on the LWGAN model:

Assumption 2. (a) Θ is a compact set. (b) The function $L(x,z;\theta)$ is continuously differentiable on Θ for all (x,z) with

$$\mathbb{E}_{X\otimes Z_0}\left[\sup_{\theta\in\Theta}\left\|\frac{\partial}{\partial\theta}L(X,A_sZ_0;\theta)\right\|^2\right] < \infty,\ s = 1,\ldots,d.$$

The compact parameter space assumption simplifies the asymptotic analysis. The moment condition rules out degenerate cases, and the differentiability is a common requirement for GAN training as various gradient descent–ascent algorithms are used. Then we adopt the ideas from Meitz (2024) to prove the estimation consistency of LWGAN.

Theorem 4. Suppose that τ_n is a sequence of nonnegative random variables such that $\tau_n \xrightarrow{P} 0$ and $n^{-1/2}/\tau_n \xrightarrow{P} 0$. Then for a fixed A, under Assumption 2, $d_H(\hat{\Theta}^*_{n,A}(\tau_n), \Theta^*_A) \xrightarrow{P} 0$ as $n \to \infty$.

Theorem 4 assures that the encoder, generator, and critic estimators of LWGAN are consistent under the Hausdorff distance for a fixed latent dimension.

4.3. Intrinsic Dimension Consistency

Finally, we show that the estimator $\hat{r}$ computed from (6) is capable of recovering the intrinsic dimension of P_X. To this end, we need to further assume that the neural network function space $\mathcal{G} \times \mathcal{Q} \times \mathcal{F}$ is large enough to cover some optimal points of interest. Define $\mathfrak{F}_A(G, Q) = \sup_{f \in \mathcal{F}^\diamond} \mathfrak{L}_A(G, Q, f)$, and let $\mathcal{G}^\diamond$ denote the set of continuous generators. Then the optimal solution set of minimizing $\overline{W}_1(P_X, P_{G(AZ_0)})$ can be characterized as

$$\mathcal{S}_A = \left\{ (G^*, Q^*, f^*) : \mathfrak{F}_A(G^*, Q^*) = \inf_{\substack{Q \in \mathcal{Q}^\diamond \\ G \in \mathcal{G}^\diamond}} \mathfrak{F}_A(G, Q), \right.$$
$$\left. \mathfrak{L}_A(G^*, Q^*, f^*) = \sup_{f \in \mathcal{F}^\diamond} \mathfrak{L}_A(G^*, Q^*, f) \right\}.$$

Clearly, coupled with some $f^\diamond \in \mathcal{F}^\diamond$, we have $(G^\diamond, Q^\diamond, f^\diamond) \in \mathcal{S}_{A_r}$. We then make the following assumption.

Assumption 3. (a) $\mathcal{S}_{A_r} \cap (\mathcal{G} \times \mathcal{Q} \times \mathcal{F}) \neq \varnothing$. (b) For each $s < r$, there exists a triplet $(G^*_s, Q^*_s, f^*_s) \in \mathcal{S}_{A_s}$ such that $f^*_s \in \mathcal{F}$ and

$$\sup_{f \in \mathcal{F}} \mathfrak{L}_{A_s}(G^*_s, Q^*_s, f) = \inf_{\substack{Q \in \mathcal{Q} \\ G \in \mathcal{G}}} \sup_{f \in \mathcal{F}} \mathfrak{L}_{A_s}(G, Q, f).$$

Now we are ready to show that the rank estimated from (6) approaches the intrinsic dimension of $\mathcal{X}$ as the sample size grows.

Theorem 5. Assume that Assumptions 2 and 3 hold. Then with $\lambda_n \to 0$ and $n^{1/2}\lambda_n \to \infty$, we have $P(\hat{r} = r) \to 1$, where $r = \texttt{InDim}(P_X)$ stands for the intrinsic dimension of $\mathcal{X}$.

Theorem 5 can be compared to the well-known Bayesian information criterion (BIC) for model selection of the following form:

$$n^{-1}\text{BIC} = -\frac{2}{n} L(\hat{\theta}; X_1, \ldots, X_n) + \frac{\log(n)}{n} \cdot s, \qquad (11)$$

where $L(\hat{\theta}; X_1, \ldots, X_n) = \sum_{i=1}^n \log p(X_i; \hat{\theta})$ is the maximized likelihood function of the model $p(x; \theta)$, $\hat{\theta}$ is the maximum likelihood estimator, and s is the number of parameters. We normalize BIC by n in (11) to make the first term comparable to an expectation.

To some extent, LWGAN and BIC share perceptible similarities. For example, if we interpret the rank s as the complexity of the model, then both LWGAN and BIC construct a penalty term $\lambda_n \cdot s$ with $\lambda_n \to 0$. More importantly, they both promise some type of model selection consistency. However, there are some fundamental differences between LWGAN and BIC. First, the theoretical rates are different. BIC has $\lambda_n = \log(n)/n$, whereas in LWGAN we require $\lambda_n \to 0$ and $n^{1/2}\lambda_n \to \infty$. Second, BIC is mostly a likelihood-based criterion, whereas in LWGAN, the main part is based on the $\overline{W}_1$ distance given in (4). Third, in the BIC framework, s always represents the number of parameters, but in LWGAN, this quantity is not meaningful, as neural networks are known to be highly overparameterized.

5. Experimental Results

In this section, we conduct comprehensive numerical experiments to validate that LWGAN is able to achieve our three goals simultaneously: detecting the correct intrinsic dimension, generating high-quality samples, and obtaining small reconstruction errors. The programming code to reproduce the experiment results is available at *https://github.com/yixuan/LWGAN*.

5.1. Simulated Experiments

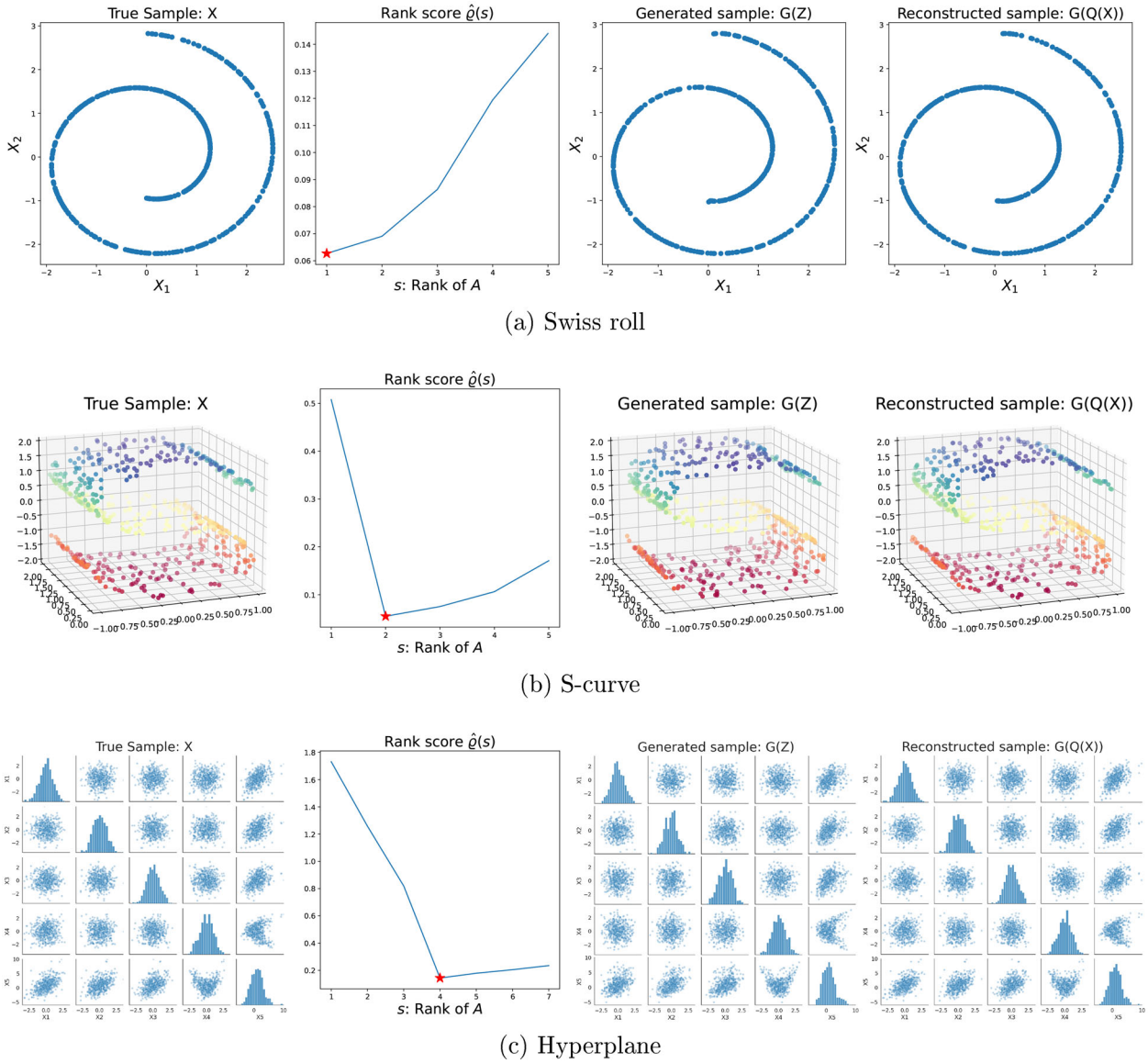
We first verify our method using three toy examples supported on manifolds with increasing dimensions. Besides the S-curve data introduced in Section 2, the other two datasets are generated as

1. Swiss roll: $X_1 = V\cos(V)$, $X_2 = V\sin(V)$, where $V = 3\pi(1 + 2U)/2$, $U \sim N(0, 1)$.
2. Hyperplane: $X_1, X_2, X_3, X_4 \overset{\text{iid}}{\sim} N(0, 1)$, $X_5 = X_1 + X_2 + X_3 + X_4^2$.

The scatterplots for the three datasets are shown in the first column of Figure 2. It is straightforward to find that the intrinsic dimensions of the Swiss roll, S-curve, and Hyperplane datasets are one, two, and four, respectively.

We then use Algorithm 1 to estimate the encoder Q and generator G for each dataset. The gradient penalty parameter is fixed to $\lambda_{\text{GP}} = 5$, and the rank regularization parameter is chosen using the method introduced in Section 3.4. After each model is trained to convergence, we compute the rank scores $\hat{\varrho}_n(s)$ defined in (7) for each s, and their values are plotted in the second column of Figure 2. From the plots we can find that the minimizers of $\hat{\varrho}_n(s)$ are consistent with the corresponding true intrinsic dimensions, which validate that LWGAN can detect the manifold dimensions of the data distributions. In Section S2.4 of the supplementary material, we also design a bootstrap-type experiment to quantify the uncertainty of the estimation results.

In addition, the third and fourth columns of Figure 2 demonstrate the model-generated points $G(Z) \equiv G(AZ_0)$ and autoencoder-reconstructed data $G(Q(X))$, respectively. Clearly, all of the plots show a high quality of the generated distribution $P_{G(Z)}$ and a small reconstruction error $\|X - G(Q(X))\|$.



(a) Swiss roll

(b) S-curve

(c) Hyperplane

Figure 2. Simulated data supported on manifolds and the demonstrations of the fitted LWAGN models.

5.2. MNIST

MNIST (LeCun et al. 1998) is a large dataset of handwritten 0–9 digits commonly used for training various image processing systems. The training set of MNIST contains 60,000 images, each consisting of 28 × 28 gray-scale pixels. It was shown that different digits have different intrinsic dimensions (Costa and Hero 2006), so the distribution of MNIST data may be supported on several disconnected manifolds with various intrinsic dimensions.

We first train models on digits 1 and 2 separately using a 16-dimensional latent variable, and the gradient penalty parameter is fixed to $\lambda_{GP} = 5$. The true sample, estimated rank scores, generated sample, and reconstructed sample for each digit are given in Figure 3. The rank score plots show that our estimation of the intrinsic dimension of digit 1 is 8, whereas the estimation of digit 2 is 12. These estimates are consistent with those of Costa and Hero (2006), which states that digit 1 exhibits a dimension estimate between 9 and 10, and digit 2 has a dimension estimate between 12 and 14.

We further estimate the intrinsic dimension of all digits from MNIST, using a similar training scheme and parameter setting, except that the maximum latent dimension is set to 20. The results for the common tasks same as above are shown in Figure 4, which suggest that the intrinsic dimension of all digits is around 16. Moreover, we also test the interpolation between two digits in the latent space. In particular, we sample pairs of testing images x_1 and x_2, and project them onto the latent space using the encoder Q, obtaining latent representations $z_1 = Q(x_1)$ and $z_2 = Q(x_2)$. We then linearly interpolate between z_1 and z_2, and pass the intermediary points through the generator G to visualize the observation-space interpolations. The results are also displayed in Figure 4, which suggest that our model can get rid of mode collapsing issues.

5.3. CelebA

CelebA (Liu et al. 2015) is another benchmark dataset for training models to generate synthetic images. It is a large-scale face

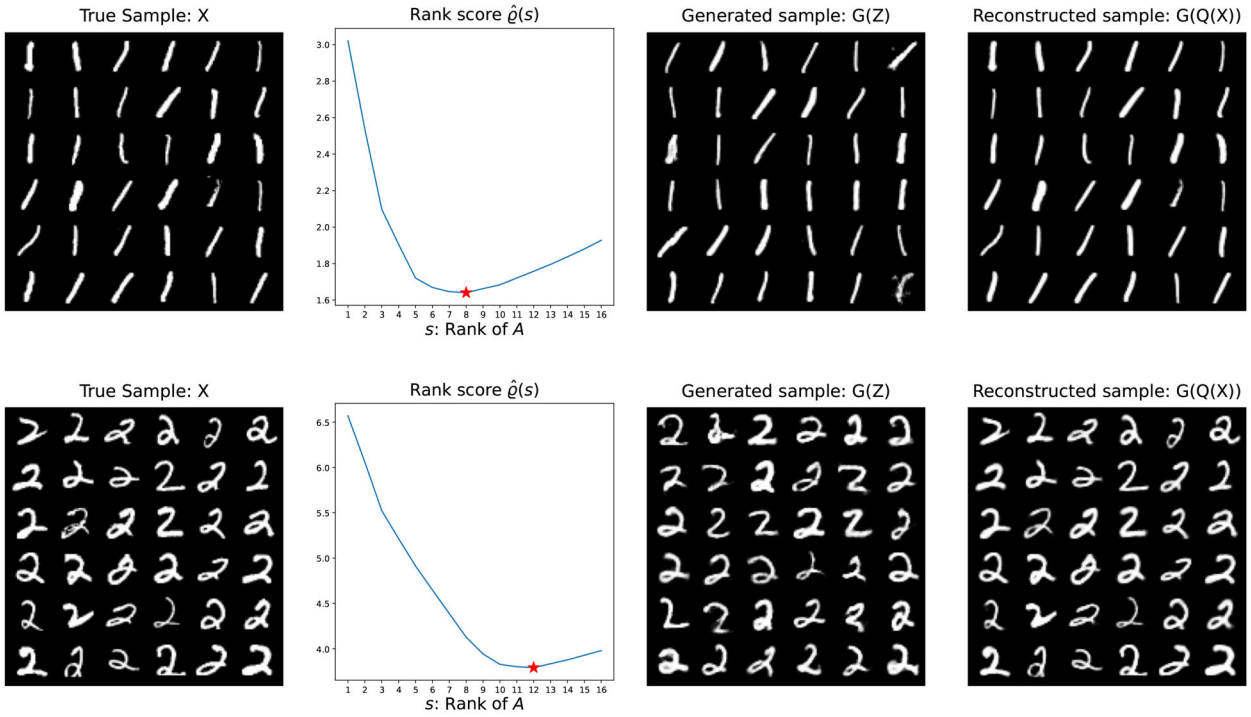


Figure 3. Digits 1 (top row) and 2 (bottom row) of the MNIST data, and the demonstrations of the fitted LWAGN models.

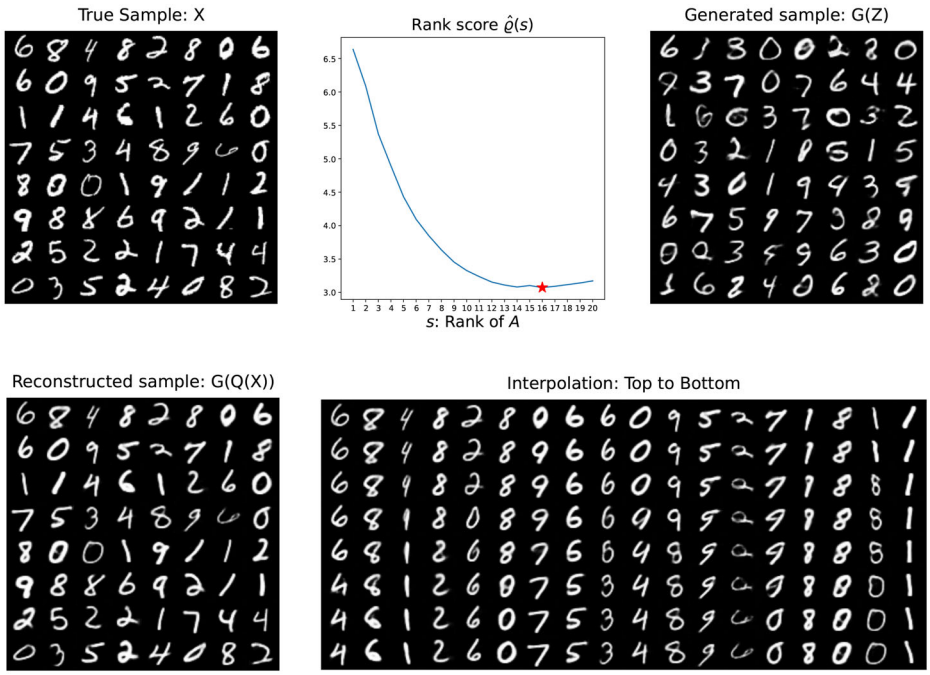
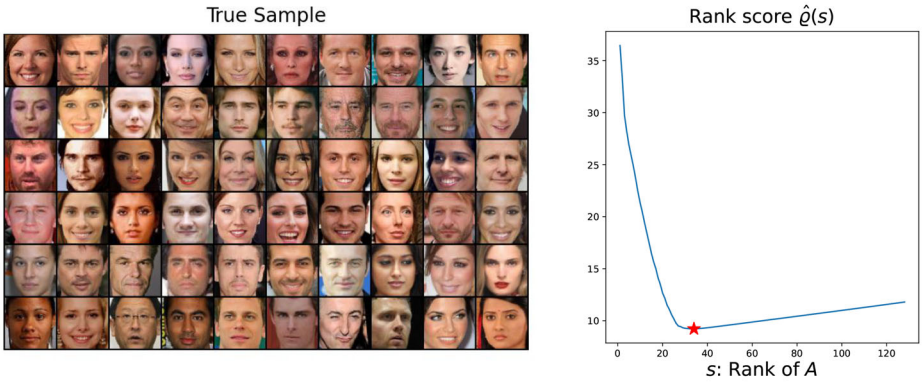


Figure 4. MNIST data with all digits and the demonstrations of the fitted LWAGN model.

attributes dataset with 202,599 color celebrity face images, which cover large pose variations. We preprocess the data by detecting the bounding box of face region in each image, cropping images to the bounding boxes, and resizing each image to 64×64 pixels. The preprocessing step has the effect of aligning the face region of each image, after which we obtain a sample of 16,055 aligned face images. A demonstration of the preprocessed CelebA images is shown in Figure 5(a).

We train CelebA using a latent dimension $d = 128$, and the rank score plot in Figure 5(b) shows that the estimated intrinsic dimension is 34. We then compare LWGAN with other generative models including WGAN, WAE, and CycleGAN (Zhu et al. 2017) both visually and numerically. In particular, the CycleGAN model introduces a cycle consistency loss based on the ℓ_1-norm to push $G(Q(X)) \approx X$ and $Q(G(Z)) \approx Z$.



(a) True sample (b) Rank scores

Figure 5. True sample of the preprocessed CelebA dataset and the rank score plot to estimate the intrinsic dimension.

Figure 6. Generated images of WGAN, WAE, CycleGAN, and LWGAN trained from the CelebA dataset.

The generated images from the four models are demonstrated in Figure 6. For LWGAN, the images are generated as $G(A_s Z_0)$, $Z_0 \sim N(0, I_d)$, where we consider different ranks $s = 16, 34, 128$. The other three methods generate images as $G(Z)$, $Z \sim N(0, I_d)$. We show the reconstructed images $G(Q(X))$ in Figure 7, and demonstrate the interpolation results in Figure 8. For these two tasks we exclude WGAN, since it does not have an encoder.

Figures 6 and 7 show that LWGAN is able to generate high-quality images as long as the rank of A_s is larger than or equal to the intrinsic dimension, and an insufficient rank results in a low quality. This validates our claims in Theorem 1 and Corollary 1. The generated images from the other three models have different levels of blur and distortion, especially for WAE. In Figure 7, we find that WAE has a good reconstruction quality, so its low generation quality may be due to the dimension mismatch between $P_{Q(X)}$ and P_Z. On the other hand, CycleGAN has a better generation quality than WAE, but it has a large reconstruction error. As a result, its reconstructed images are blurry, and it also loses many details in the interpolated images.

Finally, we numerically compare these methods with respect to three metrics: the inception scores (IS, Salimans et al. 2016), the Fréchet inception distances (FID, Heusel et al. 2017), and the reconstruction errors. IS uses a pre-trained Inception-v3 model to predict the class probabilities for each generated image, and FID improves IS by directly comparing the statistics of generated samples to real samples. For IS, higher scores are better, and for FID, lower is better. The reconstruction error is used to evaluate whether the model generates meaningful latent codes and has the capacity to recover the original information. The detailed

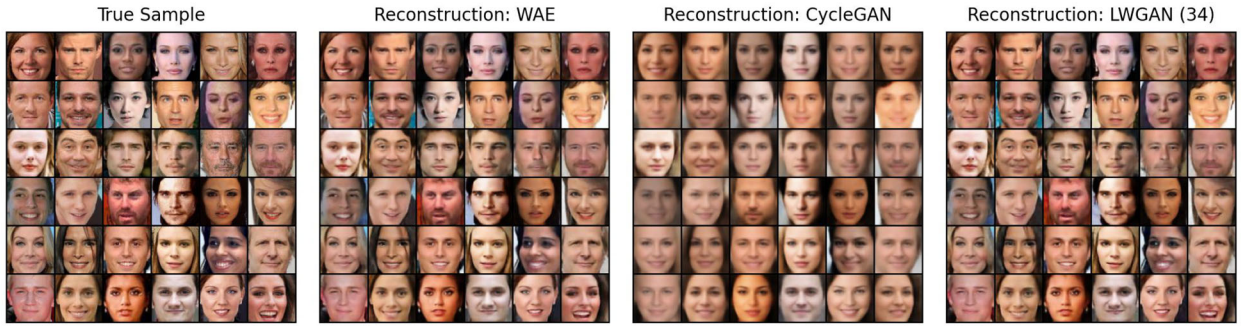


Figure 7. Reconstructed images of CelebA dataset.

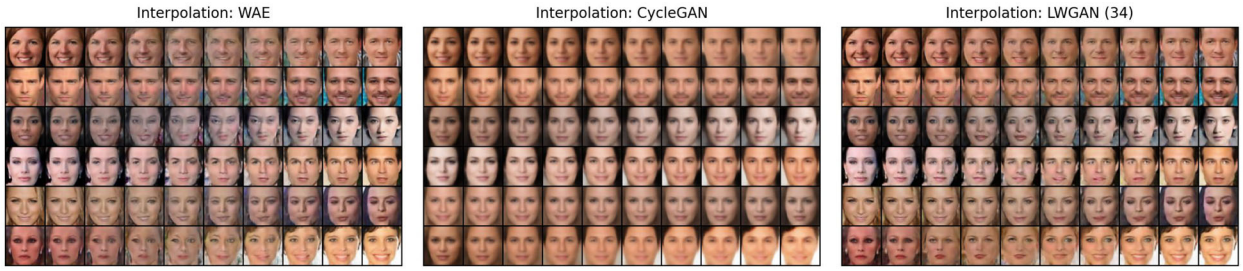


Figure 8. Interpolation of CelebA dataset.

Table 1. Numerical comparison of LWGAN, CycleGAN, WAE, and WGAN.

Methods	IS ↑	FID ↓	Reconstruction error ↓
True	2.07 (0.04)	2.77	–
LWGAN, $s = 16$	1.62 (0.02)	40.98	14.95 (3.59)
LWGAN, $s = \hat{r} = 34$	1.66 (0.03)	32.79	8.19 (1.54)
LWGAN, $s = 64$	1.70 (0.03)	**31.21**	8.15 (1.54)
LWGAN, $s = 128$	**1.71 (0.03)**	31.56	8.15 (1.54)
CycleGAN	1.54 (0.02)	42.76	20.73 (4.40)
WAE	1.59 (0.04)	51.10	**7.53 (1.35)**
WGAN	1.50 (0.03)	31.60	–

NOTE: The values in the parentheses are standard deviations.

descriptions of these three metrics are provided in Section S2.2 of the supplementary material.

Table 1 shows the values of these metrics on each trained model, with numbers in bold fonts indicating the best model for the corresponding metric, excluding the ground truth. The numerical results are consistent with our qualitative findings in Figures 6–8. Specifically, WGAN and LWGAN have relatively higher generation quality than the other two models, measured by IS and FID. WAE has a small reconstruction error, but its generation quality is low. On the contrary, CycleGAN has moderate generation quality but large reconstruction errors. For LWGAN, an insufficient rank s results in poor generation and reconstruction quality, but models with ranks larger than $\hat{r} = 34$ have good overall performance. We can also find that with the estimated rank $s = \hat{r} = 34$, LWGAN can achieve similar performance as the case of $s = d = 128$, but choosing s to be the intrinsic dimension can greatly reduce the model complexity without sacrificing the model accuracy. Overall, the proposed LWGAN is able to produce meaningful latent code and generate high-quality images at the same time, and it is the only one among all the methods compared that is capable of detecting the intrinsic dimension of data distributions.

6. Conclusion

We have developed a novel LWGAN framework that enables us to adaptively learn the intrinsic dimension of data distributions supported on manifolds. This framework fuses WAE and WGAN in a principled way, so that the model learns a latent normal distribution whose rank is consistent with the dimension of the data manifold. We have provided theoretical guarantees on the generalization error bound, estimation consistency, and dimension consistency of LWGAN. Numerical experiments have shown that the intrinsic dimension of the data can be successfully detected under several settings on both synthetic datasets and benchmark datasets, and the model-generated samples are of high quality.

A potential future direction of LWGAN is to investigate a more general scenario where the generator G is stochastic. This can be achieved by adding an extra noise vector to the input of G. In addition, it is interesting to incorporate the stochastic LWGAN into some more recent GAN modules such as BigGAN (Brock, Donahue, and Simonyan 2019), so that high-resolution and high-fidelity images can be produced along with the estimation of the intrinsic dimension.

The new LWGAN framework has many potential applications in other fields. For example, LWGAN can be used for structural estimation, which is a useful tool to quantify economic mechanisms and learn about the effects of policies that are yet to be implemented (Wei and Jiang 2022). An economic structural model specifies some outcome $g(x, \varepsilon; \theta)$ that depends on a set of observables x, unobservables ε, and structural parameters θ. 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 maximum likelihood estimator and generalized method of moments infeasible, and other simulation-based methods can cause additional computational burden. By training LWGAN on

the data from (x, y), we are able to adaptively learn the data representation by the encoder Q, instead of using moments. At the same time, we are able to boost the sample size by the generator G. By comparing the generated data $(x, g(x, \varepsilon; \theta))$ and the observed data (x, y) in the latent space, we can estimate θ efficiently.

Supplementary Materials

The supplementary materials include additional experiment details and the proofs of theorems in the main article.

Acknowledgments

The authors would like to thank the editor, the associate editor, and three reviewers for their insightful and constructive comments that have greatly helped improve this article.

Disclosure Statement

The authors report there are no competing interests to declare.

Funding

Xiao Wang's research was supported in part by the NSF Grant SES-2316428. Yixuan Qiu's work was supported in part by National Natural Science Foundation of China (12101389), Shanghai Pujiang Program (21PJC056), MOE Project of Key Research Institute of Humanities and Social Sciences (22JJD110001), and Shanghai Research Center for Data Science and Decision Technology.

References

Arjovsky, M., Chintala, S., and Bottou, L. (2017), "Wasserstein Generative Adversarial Networks," in *International Conference on Machine Learning*, pp. 214–223. [1254,1255]

Arora, S., Ge, R., Liang, Y., Ma, T., and Zhang, Y. (2017), "Generalization and Equilibrium in Generative Adversarial Nets (GANs)," in *International Conference on Machine Learning*, pp. 224–232. [1259]

Brock, A., Donahue, J., and Simonyan, K. (2019), "Large Scale GAN Training for High Fidelity Natural Image Synthesis," in *International Conference on Learning Representations*. [1265]

Chen, Y., Gao, Q., and Wang, X. (2021), "Inferential Wasserstein Generative Adversarial Networks," *Journal of the Royal Statistical Society*, Series B, 84, 83–113. [1255]

Costa, J. A., and Hero, A. O. (2006), "Determining Intrinsic Dimension and Entropy of High-Dimensional Shape Spaces," in *Statistics and Analysis of Shapes*, eds. H. Krim, and A. Yezzi, pp. 231–252, Boston: Springer. [1262]

Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2016), "Density Estimation Using Real NVP," arXiv preprint arXiv:1605.08803. [1254,1255]

Donahue, J., Krähenbühl, P., and Darrell, T. (2017), "Adversarial Feature Learning," in *International Conference on Learning Representations*. [1255]

Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M., and Courville, A. (2017), "Adversarially Learned Inference," in *International Conference on Learning Representations*. [1255]

Gao, Q., and Wang, X. (2021), "Theoretical Investigation of Generalization Bounds for Adversarial Learning of Deep Neural Networks," *Journal of Statistical Theory and Practice*, 15, 1–28. [1260]

Gao, R., Nijkamp, E., Kingma, D. P., Xu, Z., Dai, A. M., and Wu, Y. N. (2020), "Flow Contrastive Estimation of Energy-based Models," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 518–7528. [1254]

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014), "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems*, pp. 2672–2680. [1254]

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017), "Improved Training of Wasserstein GANs," in *Advances in Neural Information Processing Systems*, pp. 5767–5777. [1255,1259]

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017), "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium," in *Advances in Neural Information Processing Systems*, pp. 6626–6637. [1264]

Kingma, D. P., and Ba, J. (2014), "Adam: A Method for Stochastic Optimization," arXiv preprint arXiv:1412.6980. [1259]

Kingma, D. P., and Welling, M. (2014), "Auto-Encoding Variational Bayes," in *International Conference on Learning Representations*. [1254]

Larsen, A. B. L., Sønderby, S. K., Larochelle, H., and Winther, O. (2016), "Autoencoding Beyond Pixels Using a Learned Similarity Metric," in *International Conference on Machine Learning*, pp. 1558–1566. [1255]

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998), "Gradient-based Learning Applied to Document Recognition," *Proceedings of the IEEE*, 86, 2278–2324. [1262]

Lee, J. M. (2013), *Introduction to Smooth Manifolds*, New York: Springer. [1256]

Li, Y., Swersky, K., and Zemel, R. (2015), "Generative Moment Matching Networks," in *International Conference on Machine Learning*, pp. 1718–1727. [1254]

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015), "Deep Learning Face Attributes in the Wild," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3730–3738. [1262]

Meitz, M. (2024), "Statistical Inference for Generative Adversarial Networks and other Minimax Problems," *Scandinavian Journal of Statistics*, 51, 1323–1356. [1261]

Qiu, Y., and Wang, X. (2021), "ALMOND: Adaptive Latent Modeling and Optimization via Neural Networks and Langevin Diffusion," *Journal of the American Statistical Association*, 116, 1224–1236. [1254]

Rockafellar, R. T., and Wets, R. J.-B. (2009), *Variational Analysis*, Berlin: Springer. [1260]

Rubenstein, P. K., Schoelkopf, B., and Tolstikhin, I. (2018), "On the Latent Space of Wasserstein Auto-Encoders," arXiv preprint arXiv:1802.03761. [1255]

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016), "Improved Techniques for Training GANs," in *Advances in Neural Information Processing Systems*, pp. 2234–2242. [1264]

Tolstikhin, I., Bousquet, O., Gelly, S., and Schoelkopf, B. (2018), "Wasserstein Auto-Encoders," in *International Conference on Learning Representations*. [1255]

Villani, C. (2008), *Optimal Transport: Old and New*, Berlin: Springer. [1255]

Wei, Y., and Jiang, Z. (2022), "Estimating Parameters of Structural Models Using Neural Networks," *USC Marshall School of Business Research Paper*. [1265]

Zhang, M., Sun, Y., Zhang, C., and Mcdonagh, S. (2023), "Spread Flows for Manifold Modelling," in *International Conference on Artificial Intelligence and Statistics*, pp. 11435–11456. [1255,1257]

Zhu, J. Y., Park, T., Isola, P., and Efros, A. A. (2017), "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," in *IEEE International Conference on Computer Vision*, pp. 2223–2232. [1263]