

# Wasserstein GAN Can Perform PCA

Jaewoong Cho and Changho Suh  
School of Electrical Engineering  
Korea Advanced Institute of Science and Technology  
Email: {cjw2525,chsuh}@kaist.ac.kr

**Abstract**—Generative Adversarial Networks (GANs) have become a powerful framework to learn generative models that arise across a wide variety of domains. While there has been a recent surge in the development of numerous GAN architectures with distinct optimization metrics, we are still lacking in our understanding on how far away such GANs are from optimality. In this paper, we make progress on a theoretical understanding of the GANs under a simple linear-generator Gaussian-data setting where the optimal maximum-likelihood generator is known to perform Principal Component Analysis (PCA). We find that the original GAN by Goodfellow *et al.* fails to recover the optimal PCA solution. On the other hand, we show that Wasserstein GAN can approach the PCA solution in the limit of sample size, and hence it may serve as a basis for an optimal GAN architecture that yields the optimal generator for a wide range of data settings.

## I. INTRODUCTION

The problem of learning the probability distribution of data is one of the most fundamental problems in statistics and machine learning. A generative model plays a crucial role as an underlying framework by providing a functional block (called the generator in the literature) which can create fake data which resembles the distribution of real data. Generative Adversarial Networks (GANs) [1] have provided very powerful and efficient solutions for learning the generative model. The GAN framework includes two major components: Generator and Discriminator. This is inspired by a two-player game in which one player, Generator, wishes to generate fake samples that are close to real data, while the other player, Discriminator, wants to discriminate real samples against fake ones. Since the firstly introduced GAN [1] (that we call vanilla GAN), there has been a proliferation of GAN architectures with distinct optimization metrics including f-GAN [2], MMD-GAN [3], [4], WGAN [5], Least-Squares GAN [6], Boundary equilibrium GAN [7], etc. One natural question that arises in this context is: Is there an optimal GAN architecture among such GANs that provides optimal solutions for learning true distributions? More specifically, is there a proper optimization metric among those employed in such GANs that yields a generated distribution which maximizes the likelihood function?

In an effort to make progress towards answering this question, we take into consideration a simple canonical setting in which the optimal solution for learning distributions is well-known and hence one can readily figure out an optimization metric (if any) that yields such a solution. The simple setting represents the case in which the data has a high-dimensional Gaussian distribution and a generator is subject to

a linear operation with a Gaussian input. It has been shown in this benchmark setting that the maximum-likelihood solution performs Principal Component Analysis (PCA) [8], i.e., the covariance matrix of the generated distribution takes principal components of the true covariance matrix.

The first finding of this work is that vanilla GAN does not recover the PCA solution under the Gaussian setting when there is no constraint in the discriminator. In the discriminator-unconstrained setting, vanilla GAN can be translated into an optimization that minimizes Jensen-Shannon (JS) divergence between the true and generated distributions. Here we find that whenever the rank of the covariance matrix of the generated Gaussian distribution is smaller than that of the true distribution (which is a typical scenario), the JS divergence attains the same value of  $\log 2$  regardless of the generator [5]. Hence, vanilla GAN fails to achieve the PCA solution<sup>1</sup>.

Recently such issue on the JS divergence motivated Arjovsky *et al.* [5] to propose Wasserstein GAN (WGAN) which replaces the JS divergence with the first-order Wasserstein distance that does not pose the issue. Then, one may wonder if WGAN can recover the PCA solution in the linear Gaussian setting at hand? The main contribution of this paper is to show that this is the case. Specifically we prove that WGAN performs PCA in the limit of sample size. To prove this, we first translate WGAN optimization into another equivalent optimization. Exploiting the key condition that the optimal solution of the translated optimization should satisfy [11], we show that the optimal solution enables PCA. Our finding suggests that WGAN may be a good candidate for an optimal GAN architecture that yields an optimal generator for a wide variety of settings beyond the Gaussian case.

We also investigate stability issues for two prominent neural-net-based algorithms [5], [12] that intend to achieve the Nash equilibrium promised by the WGAN optimization. The WGAN optimization can be formulated as a minimax optimization which is actually a challenging problem as the convergence to a bad local optima may often occur. We show via empirical results that the two algorithms are actually practically appealing at least for the Gaussian setting. Our empirical simulation reveals that the deep-learning-based algorithms ensure the fast convergence to the Nash equilibrium with a small gap to the optimality.

<sup>1</sup>In practice, the discriminator has some constraints (e.g., being subject to a neural-net function). It has been studied in [9], [10] that such constraint gives a positive effect in the generator design, thus leading to a reasonably good performance in practice.

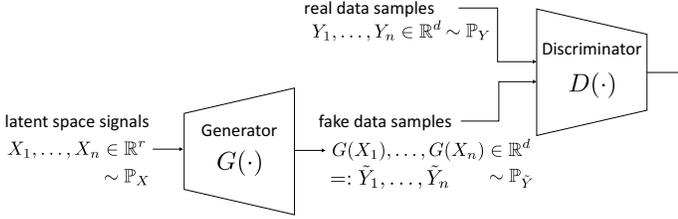


Fig. 1. An architecture for Generative Adversarial Networks (GANs).

**Related work:** Recently, Feizi *et al.* [13] explored a natural way of specifying a loss function that leads to a unified GAN architecture. The authors have investigated a quadratic loss based on the second-order Wasserstein distance to show that the corresponding GAN architecture (which they call Quadratic GAN) can perform PCA under the linear Gaussian setting. This suggests that Quadratic GAN may also be a good candidate for an optimal architecture. However, their algorithm for implementing Quadratic GAN is tailored for the Gaussian setting, so the development of generic algorithms that span a wide spectrum of data settings has been out of reach. On the other hand, we focus on WGAN for which generic neural-net-based algorithms have been well established, and promise that WGAN may be more practically appealing towards an optimal GAN architecture.

## II. GAN ARCHITECTURE

Fig. 1 illustrates an architecture for Generative Adversarial Networks (GANs). Suppose we are given the number  $n$  of real data samples. We denote those by  $(Y_1, Y_2, \dots, Y_n)$  where  $Y_i \in \mathbb{R}^d$  and  $d$  indicates a dimension of each sample. Let  $\mathbb{P}_Y$  denote the probability distribution of  $Y_i$ . Let  $(X_1, X_2, \dots, X_n)$  be latent-space signals: inputs to the generator  $G(\cdot)$  where  $X_i \in \mathbb{R}^r$ , and  $r$  indicates a dimension of the latent signal, usually  $r \leq d$ . Let  $\tilde{Y}_i := G(X_i) \in \mathbb{R}^d$  be the  $i$ th fake sample and  $\mathbb{P}_{\tilde{Y}}$  be the probability distribution of  $\tilde{Y}_i$ .

We focus on a linear-generator Gaussian-data setting in which  $Y_i$ 's are i.i.d. each according to  $\mathbb{P}_Y = \mathcal{N}(\mathbf{0}, \mathbf{K}_Y)$ ;  $X_i$ 's are i.i.d.  $\sim \mathcal{N}(\mathbf{0}, \mathbf{I}_r)$ ; and  $G(\cdot)$  is a linear operator implemented by a matrix  $\mathbf{G} \in \mathbb{R}^{d \times r}$ .

The GAN architecture is inspired by a two-player game in which one player is the discriminator  $D(\cdot)$  who wishes to discriminate real samples against fake ones; and the other is the generator  $G(\cdot)$  who wants to fool the discriminator. Many GAN approaches have been developed as an effort to achieve such goals, and corresponding optimization problems are formulated with different optimization metrics [1]–[7].

## III. VANILLA GAN

In an effort to achieve such goals, Goodfellow *et al.* [1] came up with an insightful interpretation: Viewing  $D(\cdot)$  as the *probability* that the input (taking either a real sample  $Y_i$  or a fake one  $\tilde{Y}_i$ ) is a real sample. This viewpoint motivated

them to introduce the following minimax optimization. Given  $(X_1, Y_1), \dots, (X_n, Y_n)$ :

$$\min_{G(\cdot)} \max_{D(\cdot)} \frac{1}{n} \sum_{i=1}^n \log D(Y_i) + \frac{1}{n} \sum_{i=1}^n \log(1 - D(\tilde{Y}_i)).$$

Notice that the discriminator intends to maximize the logarithmic of such probability  $D(\cdot)$  when the input is real (or the logarithm of “1 – such probability”  $1 - D(\cdot)$  for a fake sample).

For simplicity, we consider a regime in which the sample size  $n$  is large enough that the objective function in the above can be well approximated as the population limit. So the optimization can be approximated as:

$$\min_{G(\cdot)} \max_{D(\cdot)} \mathbb{E}_Y[\log D(Y)] + \mathbb{E}_{\tilde{Y}}[\log(1 - D(\tilde{Y}))]. \quad (1)$$

In this work, we find that vanilla GAN does not recover the optimal PCA solution even for the simple Gaussian setting. To see this, we first rewrite the objective function as:

$$\int_{z \in \mathcal{Y} \cup \tilde{\mathcal{Y}}} [\mathbb{P}_Y(z) \log D(z) + \mathbb{P}_{\tilde{Y}}(z) \log(1 - D(z))] dz, \quad (2)$$

where  $\mathcal{Y}$  and  $\tilde{\mathcal{Y}}$  indicate the ranges of  $Y$  and  $\tilde{Y}$ , respectively. Here we set:

$$\begin{aligned} \mathbb{P}_Y(z) dz &:= 0 & \text{if } z \in \tilde{\mathcal{Y}} \setminus \mathcal{Y}; \\ \mathbb{P}_{\tilde{Y}}(z) dz &:= 0 & \text{if } z \in \mathcal{Y} \setminus \tilde{\mathcal{Y}}. \end{aligned}$$

Observe that for any  $(a, b) \in \mathbb{R}^2 \setminus \{0, 0\}$  and  $t \in (0, 1)$ , the function  $a \log t + b \log(1 - t)$  is maximized at  $t^* = \frac{a}{a+b}$ . Hence, for a fixed  $G(\cdot)$ , the optimal discriminator reads:

$$D^*(z) = \frac{\mathbb{P}_Y(z)}{\mathbb{P}_Y(z) + \mathbb{P}_{\tilde{Y}}(z)}.$$

Plugging this into (1), we get:

$$\begin{aligned} & \min_{\mathbf{G}} \mathbb{E}_Y[\log D^*(Y)] + \mathbb{E}_{\tilde{Y}}[\log(1 - D^*(\tilde{Y}))] \\ &= \min_{\mathbf{G}} \mathbb{E}_Y \left[ \log \frac{\mathbb{P}_Y(Y)}{\mathbb{P}_{\tilde{Y}}(Y) + \mathbb{P}_Y(Y)} \right] + \mathbb{E}_{\tilde{Y}} \left[ \log \frac{\mathbb{P}_{\tilde{Y}}(\tilde{Y})}{\mathbb{P}_Y(\tilde{Y}) + \mathbb{P}_{\tilde{Y}}(\tilde{Y})} \right] \\ &= \min_{\mathbf{G}} \text{KL} \left( \mathbb{P}_Y \left\| \frac{\mathbb{P}_Y + \mathbb{P}_{\tilde{Y}}}{2} \right\| \right) + \text{KL} \left( \mathbb{P}_{\tilde{Y}} \left\| \frac{\mathbb{P}_Y + \mathbb{P}_{\tilde{Y}}}{2} \right\| \right) - \log 4 \\ &= \min_{\mathbf{G}} 2 \cdot \text{JSD}(\mathbb{P}_Y \parallel \mathbb{P}_{\tilde{Y}}) - \log 4, \end{aligned}$$

where  $\text{KL}(\cdot \parallel \cdot)$  and  $\text{JSD}(\cdot \parallel \cdot)$  indicate the KL and JS divergences, respectively [14], [15]: For distributions  $p$  and  $q$ ,

$$\text{JSD}(p \parallel q) := \frac{1}{2} \left( \text{KL} \left( p \left\| \frac{p+q}{2} \right\| \right) + \text{KL} \left( q \left\| \frac{p+q}{2} \right\| \right) \right).$$

In the linear Gaussian setting,  $\tilde{Y}$  is also Gaussian  $\mathcal{N}(\mathbf{0}, \mathbf{K}_{\tilde{Y}})$  where  $\mathbf{K}_{\tilde{Y}} = \mathbb{E}[\mathbf{G}X(\mathbf{G}X)^T] = \mathbf{G}\mathbf{G}^T$ . Typically the dimension of the latent signals is smaller than that of data. So in this case, the support of  $\mathbb{P}_{\tilde{Y}}$  has a strictly lower dimension than that of  $\mathbb{P}_Y$ , yielding:

$$\frac{\mathbb{P}_Y(z) + \mathbb{P}_{\tilde{Y}}(z)}{2} dz = \begin{cases} \frac{\mathbb{P}_Y(z)}{2} dz & \text{if } z \in \mathcal{Y} \setminus \tilde{\mathcal{Y}}; \\ \frac{\mathbb{P}_{\tilde{Y}}(z)}{2} dz & \text{if } z \in \tilde{\mathcal{Y}} \setminus \mathcal{Y}. \end{cases} \quad (3)$$

This then gives  $\text{KL}\left(\mathbb{P}_Y \left\| \frac{\mathbb{P}_Y + \mathbb{P}_{\tilde{Y}}}{2} \right\| \right) = \text{KL}\left(\mathbb{P}_{\tilde{Y}} \left\| \frac{\mathbb{P}_Y + \mathbb{P}_{\tilde{Y}}}{2} \right\| \right) = \log 2$ , which in turn yields:

$$\text{JSD}(\mathbb{P}_Y \|\mathbb{P}_{\tilde{Y}}) = \log 2, \quad (4)$$

regardless of how we design  $\mathbf{G}$ . This implies that an optimal  $\mathbf{G}^*$  does not necessarily perform PCA.

#### IV. WASSERSTEIN GAN

The main reason that vanilla GAN fails to recover the optimal solution is that it is based on the JS divergence which poses the critical issue reflected in (4). As an effort to avoid such an issue, one may consider another prominent GAN architecture developed by Arjovsky *et. al.* [5]: Wasserstein GAN (WGAN) which employs the first-order Wasserstein distance instead of the JS divergence. The first-order Wasserstein distance is defined as: Given two distributions  $\mathbb{P}_Y$  and  $\mathbb{P}_{\tilde{Y}}$ ,

$$W(\mathbb{P}_Y, \mathbb{P}_{\tilde{Y}}) := \min_{\mathbb{P}_{Y, \tilde{Y}}} \mathbb{E} \left[ \|Y - \tilde{Y}\| \right], \quad (5)$$

where the minimization is over all joint distributions which respect the marginals  $\mathbb{P}_Y$  and  $\mathbb{P}_{\tilde{Y}}$ . Here  $\|\cdot\|$  indicates the  $\ell_2$  norm. Unlike the JS divergence, it provides a meaningful non-saturating value even when dimensions of supports of two distributions are distinct. WGAN intends to solve the following optimization:

$$\min_{\mathbf{G}} W(\mathbb{P}_Y, \mathbb{P}_{\tilde{Y}}). \quad (6)$$

The main contribution of this paper is that unlike vanilla GAN, WGAN recovers the optimal PCA solution under the linear Gaussian setting in the limit of sample size, formally stated below.

**Theorem 1.** *Let  $Y \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_Y)$  where  $\mathbf{K}_Y$  has a full rank, i.e.,  $\text{rank}(\mathbf{K}_Y) = d$ . Let  $X \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_r)$  where  $r \leq d$ . The optimal solution of the WGAN optimization (6) under a linear generator is the  $r$ -PCA solution.*

*Proof.* The proof consists of two parts:

- The WGAN optimization (6) is translated into another equivalent optimization which was investigated in depth in prior works [11];
- We show that the  $r$ -PCA solution respects the unique condition (derived in [11]) that the optimal solution of the translated optimization satisfies.

By the definition of the Wasserstein distance (5) and the assumption of a linear generator,

$$\min_{\mathbf{G}} W(\mathbb{P}_Y, \mathbb{P}_{\mathbf{G}X}) = \min_{\mathbf{G}} \min_{\mathbb{P}_{Y, \mathbf{G}X}} \mathbb{E} [\|Y - \mathbf{G}X\|]. \quad (7)$$

Lemma 1 below casts the WGAN optimization (7) into another equivalent optimization.

**Lemma 1.** *Under a linear generator,*

$$\min_{\mathbf{G}} \min_{\mathbb{P}_{Y, \mathbf{G}X}} \mathbb{E} [\|Y - \mathbf{G}X\|] = \min_{\mathbf{U}^T \mathbf{U} = \mathbf{I}_r} \mathbb{E} [\|Y - \mathbf{U}\mathbf{U}^T Y\|]. \quad (8)$$

*Proof.* See Section IV-A.

*Remark:* Lemma 1 suggests that for the optimal  $\mathbf{G}^*$  and  $\mathbf{U}^*$ , we have  $\mathbf{G}^* X = \mathbf{U}^* \mathbf{U}^{*T} Y$ . Considering the covariances of these quantities, we see that

$$\mathbf{G}^* \mathbf{G}^{*T} = \mathbf{U}^* \mathbf{U}^{*T} \mathbf{K}_Y \mathbf{U}^* \mathbf{U}^{*T}. \quad (9)$$

□

For the translated optimization (8), [11] derived the key condition that the optimal solution  $\mathbf{U}^*$  should satisfy. This condition turns to play a crucial role to prove the theorem. We first introduce a notation which serves describing the condition. Let

$$\mathbf{M} = \mathbb{E} \left[ \frac{Y Y^T}{\|Y - \mathbf{U}^* \mathbf{U}^{*T} Y\|} \right]. \quad (10)$$

Now the key condition w.r.t.  $\mathbf{U}^*$  says:

$$\mathbf{U}^* = r\text{-PrincipalEigenvectors}(\mathbf{M}), \quad (11)$$

which means a concatenation of the  $r$  principal eigenvectors of  $\mathbf{M}$ . It was also shown in [11] that  $\mathbf{U}^*$  satisfying the key condition (11) is *unique*. Here what we prove is that such  $\mathbf{U}^*$  satisfies (see below for the proof):

$$\mathbf{U}^* = r\text{-PrincipalEigenvectors}(\mathbf{K}_Y). \quad (12)$$

Applying (12) into (9), we can find the optimal  $\mathbf{G}^*$  satisfies:

$$\mathbf{G}^* \mathbf{G}^{*T} = \mathbf{V} \text{diag}(\sigma_1^2, \dots, \sigma_r^2, 0, \dots, 0) \mathbf{V}^T \quad (13)$$

where  $\mathbf{V} := [\mathbf{v}_1, \dots, \mathbf{v}_d] \in \mathbb{R}^{d \times d}$  and  $\sigma_i^2$  are eigen-components of  $\mathbf{K}_Y$ :

$$\mathbf{K}_Y = \mathbf{V} \text{diag}(\sigma_1^2, \dots, \sigma_d^2) \mathbf{V}^T =: \mathbf{V} \mathbf{\Sigma} \mathbf{V}^T. \quad (14)$$

Here  $\mathbf{v}_i$ 's are orthonormal vectors and  $\sigma_1 \geq \dots \geq \sigma_d$ . This implies that the optimal generator  $\mathbf{G}^*$  is formed by taking the  $r$  principal components of  $\mathbf{K}_Y$ . This completes the proof. From below, we will prove the main claim (12).

*Proof of (12):* Since  $\mathbf{V}$  is an orthonormal matrix, we get:

$$Y = \mathbf{V} \mathbf{V}^T Y = \sum_{i=1}^d Z_i \mathbf{v}_i,$$

where  $Z_i := \mathbf{v}_i^T Y$ . Since  $Y \sim \mathcal{N}(\mathbf{0}, \mathbf{V} \mathbf{\Sigma} \mathbf{V}^T)$  (14),

$$Z := \mathbf{V}^T Y = [Z_1; \dots; Z_d] \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}). \quad (15)$$

Let  $\mathbf{V}_r := [\mathbf{v}_1, \dots, \mathbf{v}_r]$  ( $r$ -principal eigenvectors of  $\mathbf{K}_Y$ ). Now we intend to show that  $\mathbf{V}_r = \mathbf{U}^*$ . To this end, it suffices to show that  $\mathbf{V}_r$  satisfies the key condition of (10) due to the

uniqueness of  $\mathbf{U}^*$ . So, we plug  $\mathbf{V}_r$  into (10) (by replacing  $\mathbf{U}^*$ ), thus obtaining:

$$\begin{aligned} \mathbf{M} &= \mathbb{E} \left[ \frac{\mathbf{Y}\mathbf{Y}^T}{\|\mathbf{Y} - \mathbf{V}_r \mathbf{V}_r^T \mathbf{Y}\|} \right] = \mathbb{E} \left[ \frac{\mathbf{Y}\mathbf{Y}^T}{\sqrt{\|\sum_{i=r+1}^d Z_i \mathbf{v}_i\|^2}} \right] \\ &\stackrel{(a)}{=} \mathbb{E} \left[ \frac{\mathbf{Y}\mathbf{Y}^T}{\sqrt{\sum_{i=r+1}^d Z_i^2}} \right] = \mathbb{E} \left[ \frac{\left(\sum_{j=1}^d Z_j \mathbf{v}_j\right) \left(\sum_{k=1}^d Z_k \mathbf{v}_k^T\right)}{\sqrt{\sum_{i=r+1}^d Z_i^2}} \right] \\ &= \sum_{j=1}^d \mathbb{E} \left[ \frac{Z_j^2}{\sqrt{\sum_{i=r+1}^d Z_i^2}} \right] \mathbf{v}_j \mathbf{v}_j^T \\ &\quad + \sum_{j \neq k, j < k} 2 \mathbb{E} \left[ \frac{Z_j Z_k}{\sqrt{\sum_{i=r+1}^d Z_i^2}} \right] \mathbf{v}_j \mathbf{v}_k^T \\ &\stackrel{(b)}{=} \sum_{j=1}^d \mathbb{E} \left[ \frac{Z_j^2}{\sqrt{\sum_{i=r+1}^d Z_i^2}} \right] \mathbf{v}_j \mathbf{v}_j^T, \end{aligned} \quad (16)$$

where (a) follows from the orthonormality of  $\mathbf{v}_i$ 's and (b) follows from the fact that

$$\mathbb{E} \left[ \frac{Z_j Z_k}{\sqrt{\sum_{i=r+1}^d Z_i^2}} \right] = 0, \quad (17)$$

which we will show in the sequel.

*Proof of (17):* Let  $Z^r := \{Z_{r+1}, \dots, Z_d\}$ . Using the tower property and the fact that  $Z = [Z_1; \dots; Z_d] \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$  (see (15)), we get

$$\begin{aligned} \mathbb{E} \left[ \frac{Z_j Z_k}{\sqrt{\sum_{i=r+1}^d Z_i^2}} \right] &= \mathbb{E}_{Z^r, Z_k} \left[ \mathbb{E}_{Z_j} \left[ \frac{Z_j Z_k}{\sqrt{\sum_{i=r+1}^d Z_i^2}} \middle| Z^r, Z_k \right] \right] \\ &\stackrel{(a)}{=} \mathbb{E}_{Z^r, Z_k} \left[ \mathbb{E}_{Z_j} \left[ \frac{-Z_j Z_k}{\sqrt{\sum_{i=r+1}^d Z_i^2}} \middle| Z^r, Z_k \right] \right] \\ &= -\mathbb{E} \left[ \frac{Z_j Z_k}{\sqrt{\sum_{i=r+1}^d Z_i^2}} \right], \end{aligned}$$

where (a) follows from the fact that  $Z_j \sim \mathcal{N}(0, \sigma_j^2)$  has a symmetric pdf and  $\frac{Z_j Z_k}{\sqrt{\sum_{i=r+1}^d Z_i^2}}$  is an odd function of  $Z_j$ . Therefore,

$$\mathbb{E} \left[ \frac{Z_j Z_k}{\sqrt{\sum_{i=r+1}^d Z_i^2}} \right] = 0.$$

Computing  $\mathbf{M}\mathbf{v}_k$  with the help of (16), we get:

$$\mathbf{M}\mathbf{v}_k = \mathbb{E} \left[ \frac{Z_k^2}{\sqrt{\sum_{i=r+1}^d Z_i^2}} \right] \mathbf{v}_k.$$

This implies that  $\mathbf{v}_k$  is an eigenvector of  $\mathbf{M}$  and the corresponding eigenvalue is  $\mathbb{E} \left[ \frac{Z_k^2}{\sqrt{\sum_{i=r+1}^d Z_i^2}} \right]$ .

Now it suffices to show that  $(\mathbf{v}_1, \dots, \mathbf{v}_r)$  are  $r$ -principal eigenvectors of  $\mathbf{M}$ . To show this, we will demonstrate below that for  $j \leq r$  and  $k > r$ ,

$$\mathbb{E} \left[ \frac{Z_j^2}{\sqrt{\sum_{i=r+1}^d Z_i^2}} \right] \geq \mathbb{E} \left[ \frac{Z_k^2}{\sqrt{\sum_{i=r+1}^d Z_i^2}} \right]. \quad (18)$$

Since  $j \leq r$  and  $Z_i$ 's are independent, we have:

$$\begin{aligned} \mathbb{E} \left[ \frac{Z_j^2}{\sqrt{\sum_{i=r+1}^d Z_i^2}} \right] &= \mathbb{E}[Z_j^2] \mathbb{E} \left[ \frac{1}{\sqrt{\sum_{i=r+1}^d Z_i^2}} \right] \\ &\stackrel{(a)}{\geq} \mathbb{E}[Z_k^2] \mathbb{E} \left[ \frac{1}{\sqrt{Z_k^2 + \sum_{i>r, i \neq k} Z_i^2}} \right] \\ &\stackrel{(b)}{=} \mathbb{E}_{Z_k}[Z_k^2] \mathbb{E}_{Z^r} \left[ \mathbb{E}_{Z_k} \left[ \frac{1}{\sqrt{Z_k^2 + \sum_{i>r, i \neq k} Z_i^2}} \middle| Z^r \right] \right] \\ &\stackrel{(c)}{\geq} \mathbb{E}_{Z_k}[Z_k^2] \mathbb{E}_{Z^r} \left[ \frac{1}{\sqrt{\mathbb{E}_{Z_k}[Z_k^2] + \sum_{i>r, i \neq k} Z_i^2}} \right] \\ &= \mathbb{E}_{Z^r} \left[ \frac{\mathbb{E}_{Z_k}[Z_k^2]}{\sqrt{\mathbb{E}_{Z_k}[Z_k^2] + \sum_{i>r, i \neq k} Z_i^2}} \right] \\ &\stackrel{(d)}{\geq} \mathbb{E}_{Z^r} \left[ \mathbb{E}_{Z_k} \left[ \frac{Z_k^2}{\sqrt{Z_k^2 + \sum_{i>r, i \neq k} Z_i^2}} \middle| Z^r \right] \right] \\ &= \mathbb{E} \left[ \frac{Z_k^2}{\sqrt{\sum_{i=r+1}^d Z_i^2}} \right], \end{aligned}$$

where (a) follows from  $\mathbb{E}[Z_j^2] = \sigma_j^2 \geq \sigma_k^2 = \mathbb{E}[Z_k^2]$  ( $j < k$ ); (b) follows from the tower property and  $Z \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ ; (c) follows from Jensen's inequality and the fact that  $f(x) = \frac{1}{\sqrt{x+a}}$  is a convex function of  $x \geq 0$ ; and (d) follows from Jensen's inequality and the fact that  $f(x) = \frac{x}{\sqrt{x+a}}$  is a concave function of  $x \geq 0$ . This proves (18).  $\blacksquare$

#### A. Proof of Lemma 1

Let  $\tilde{\mathbf{Y}} = \mathbf{G}\mathbf{X}$  be a random vector whose support lies in  $\mathcal{S}$  ( $r$ -dimensional subspace in  $\mathbb{R}^d$ ). Then we can write  $\mathbf{Y}$  as:

$$\mathbf{Y} = \mathbf{Y}_{\mathcal{S}'} + \mathbf{Y}_{\mathcal{S}}, \quad (19)$$

where  $\mathbf{Y}_{\mathcal{S}}$  indicates the projection of  $\mathbf{Y}$  onto  $\mathcal{S}$ . Using this,

$$\begin{aligned} \min_{\mathbf{G}} W(\mathbb{P}_Y, \mathbb{P}_{\tilde{\mathbf{Y}}}) &= \min_{\mathbf{G}} \min_{\mathbb{P}_{Y, \tilde{\mathbf{Y}}}} \mathbb{E}[\|\mathbf{Y} - \tilde{\mathbf{Y}}\|] \\ &= \min_{\mathbf{G}} \min_{\mathbb{P}_{Y, \tilde{\mathbf{Y}}}} \mathbb{E} \left[ \sqrt{\|\mathbf{Y}_{\mathcal{S}'} + \mathbf{Y}_{\mathcal{S}} - \tilde{\mathbf{Y}}\|^2} \right] \\ &\stackrel{(a)}{=} \min_{\mathbf{G}} \min_{\mathbb{P}_{Y, \tilde{\mathbf{Y}}}} \mathbb{E} \left[ \sqrt{\|\mathbf{Y}_{\mathcal{S}'}\|^2 + \|\mathbf{Y}_{\mathcal{S}} - \tilde{\mathbf{Y}}\|^2} \right], \end{aligned}$$

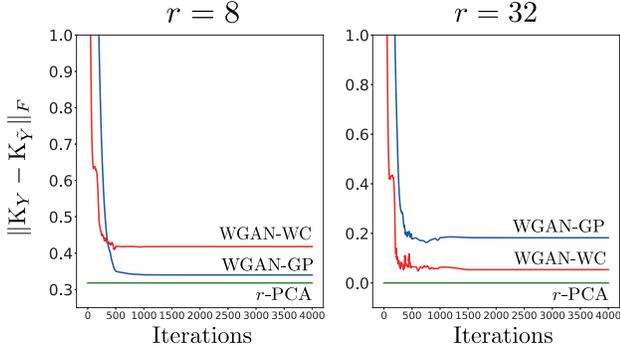


Fig. 2. The performance of WGAN-GP and WGAN-WC for the linear Gaussian setting in which the data dimension  $d = 32$  and latent signal dimension  $r \in \{8, 32\}$ . As a performance metric, we employ the Frobenius norm between the ground-truth covariance and the generated covariance. As a baseline, we plot the performance of the  $r$ -PCA solution.

where (a) follows from the orthogonality between  $Y_{S'}$  and  $(Y_S - \tilde{Y})$ . Let  $\mathbb{P}_{Y, \tilde{Y}}^*$  denote a joint distribution such that  $Y_S = \tilde{Y}$ . Then

$$\begin{aligned} & \mathbb{E}_{\mathbb{P}_{Y, \tilde{Y}}^*} \left[ \sqrt{\|Y_{S'}\|^2 + \|Y_S - \tilde{Y}\|^2} \right] \\ &= \mathbb{E}_{\mathbb{P}_Y} \left[ \sqrt{\|Y_{S'}\|^2} \right] \geq \min_{\mathbb{P}_{Y, \tilde{Y}}} \mathbb{E} \left[ \sqrt{\|Y_{S'}\|^2 + \|Y_S - \tilde{Y}\|^2} \right]. \end{aligned}$$

From the fact that  $\sqrt{\|Y_{S'}\|^2} \leq \sqrt{\|Y_{S'}\|^2 + \|Y_S - \tilde{Y}\|^2}$ ,

$$\begin{aligned} & \mathbb{E}_{\mathbb{P}_{Y, \tilde{Y}}^*} \left[ \sqrt{\|Y_{S'}\|^2 + \|Y_S - \tilde{Y}\|^2} \right] \\ &= \mathbb{E}_{\mathbb{P}_Y} \left[ \sqrt{\|Y_{S'}\|^2} \right] \leq \min_{\mathbb{P}_{Y, \tilde{Y}}} \mathbb{E} \left[ \sqrt{\|Y_{S'}\|^2 + \|Y_S - \tilde{Y}\|^2} \right]. \end{aligned}$$

Therefore,

$$\begin{aligned} & \min_{\mathbf{G}} W(\mathbb{P}_Y, \mathbb{P}_{\tilde{Y}}) \\ &= \min_{\mathbf{G}} \min_{\mathbb{P}_{Y, \tilde{Y}}} \mathbb{E} \left[ \sqrt{\|Y_{S'}\|^2 + \|Y_S - \tilde{Y}\|^2} \right] \\ &= \min_{\mathbf{G}} \mathbb{E}_{\mathbb{P}_Y} \left[ \sqrt{\|Y_{S'}\|^2} \right] = \min_S \mathbb{E}_{\mathbb{P}_Y} [\|Y - Y_S\|] \\ &= \min_{\mathbf{U}^T \mathbf{U} = \mathbf{I}_r} \mathbb{E}_{\mathbb{P}_Y} [\|Y - \mathbf{U}\mathbf{U}^T Y\|], \end{aligned}$$

where  $\text{range}(\mathbf{G}) = \text{range}(\mathbf{U})$ .

## V. NEURAL-NET-BASED WGAN ALGORITHMS

So far we have shown that WGAN recovers the optimal PCA solution for the linear Gaussian setting. This suggests that WGAN can be a good candidate for the optimal GAN architecture.

In practice, however, it is early to arrive at this conclusion. To implement the WGAN solution, usually we hinge upon the Kantorovich dual [16] of the primal optimization, which includes two function optimizations that can be efficiently solved via neural networks with high accuracy:

$$\min_{\mathbf{G}} W(\mathbb{P}_Y, \mathbb{P}_{\mathbf{G}X}) = \min_{\mathbf{G}} \max_{\|D\|_L \leq 1} \mathbb{E}_Y [D(Y)] - \mathbb{E}_{\tilde{Y}} [D(\tilde{Y})],$$

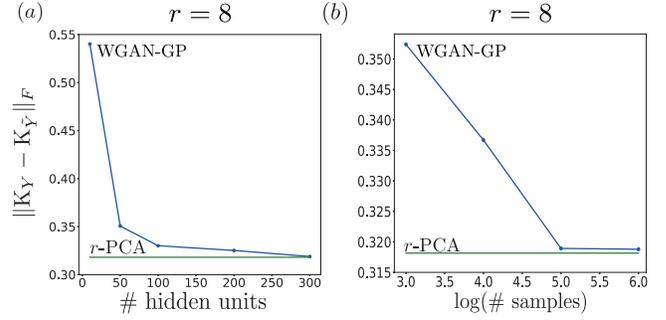


Fig. 3. The performance of WGAN-GP for the linear Gaussian setting in which the data dimension  $d = 32$  and latent signal dimension  $r = 8$  in different network sizes with different sample sizes. The discriminator of WGAN-GP is implemented with a neural network composed of five layers in which hidden layers include the same number of hidden units: (a) the performance of WGAN-GP with the sample size  $n = 100,000$  in different sizes of hidden units  $n_h \in \{10, 50, 100, 200, 300\}$ ; (b) the performance of WGAN-GP with different sample sizes  $n \in \{1000, 10000, 100000, 1000000\}$  for the case of  $n_h = 300$ . We found that the gap reduces when a more complex neural network is employed together with a larger  $n$ .

where  $\|D\|_L \leq 1$  means 1-Lipschitz functions  $D$ : For all  $y_1$  and  $y_2$ ,  $|D(y_1) - D(y_2)| \leq \|y_1 - y_2\|$ . But here an issue arises: The Kantorovich dual is of *minimax* optimization where the convergence to a bad local optima frequently occurs. Hence, we also need to investigate the stability of neural-net-based algorithms that implement the minimax solution.

In this work, we consider such two prominent algorithms: (1) WGAN with weight clipping (WGAN-WC); (2) WGAN with gradient penalty (WGAN-GP).

Another contribution of this work is to empirically show that the two algorithms yield good stability for the Gaussian setting, i.e., ensure the fast convergence to the Nash equilibrium with a small gap to the optimality.

An empirical verification is done for the following setting. We generate  $n \in \{1000, 10000, 100000, 1000000\}$  i.i.d. samples from an  $d$ -dimensional Gaussian distribution  $Y \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_Y)$ . We set  $d = 32$ . A generation for  $\mathbf{K}_Y$  takes the following procedure. First we generate  $\mathbf{V}\Sigma\mathbf{V}^T$  such that  $[\mathbf{V}]_{ij} \sim \mathcal{N}(0, 1)$  and  $\sigma_i^2 \sim \text{Uniform}(0, 10)$ . Next, we normalize  $\mathbf{V}\Sigma\mathbf{V}^T$  so as to have the unit Frobenius norm. The generator is constructed with a single matrix  $\mathbf{G}$ . The discriminator is implemented with a fully connected neural network. As a performance metric, we employ the Frobenius norm between the ground-truth covariance and the generated covariance. We use the batch size of 200. For WGAN-GP, we use Adam optimizer to train neural networks and momentum parameters  $(\beta_1, \beta_2) = (0.5, 0.9)$ , and we set  $\lambda$  parameter to be 0.1. For WGAN-WC, we use RMSProp optimizer, and we set  $c$  parameter to be 0.01.

Fig. 2 shows the performance of WGAN-GP (blue-colored) and WGAN-WC (red-colored) for the values of  $r \in \{8, 32\}$  and  $n = 100,000$  with the Glorot initialization [17]. Here the discriminator is implemented with a neural network composed of three layers, each with 64 neurons and ReLU activation functions. We use  $10^{-3}$  as the initial learning rate, and decay

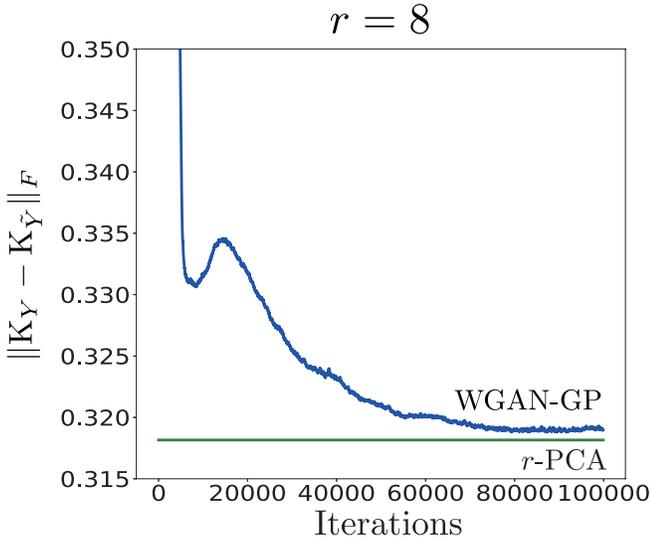


Fig. 4. The performance of WGAN-GP for the linear Gaussian setting in which the data dimension  $d = 32$  and latent signal dimension  $r = 8$  with the sample size  $n = 1,000,000$ . The discriminator is implemented with a neural network composed of five layers, each with 300 neurons and ReLU activation functions. We can observe a small gap in the convergence.

it by a factor of 10 at the end of every 5 epochs for both algorithms. For comparison, we also plot the performance of the  $r$ -PCA with a green-colored line. Observe that both WGAN-GP and WGAN-WC exhibit reasonably fast convergence speed<sup>2</sup> (converge only with a few hundred iterations) with a small gap to the optimality, reflected in the green-colored  $r$ -PCA solution. Here the small gap comes from a finite sample size  $n$ , as well as imperfect representability of a specific neural network employed herein. Actually we found that the gap reduces when a more complex neural network is employed together with a larger  $n$ . Here are details - also see Figs. 3 and 4.

As a setting that incorporates a more complex network, we consider the case which  $r = 8$ ,  $n = 100,000$ , and the number of hidden units for each layer  $n_h \in \{10, 50, 100, 200, 300\}$ ; see Fig. 3(a). Here we employ WGAN-GP. The discriminator is implemented with a neural network composed of five layers in which hidden layers include the same number of hidden units and we use Adam optimizer with the learning rate  $10^{-4}$ . We see from Fig. 3(a) that the gap reduces with an increasing complexity, converging almost to 0. Fig. 3(b) demonstrates the performance of WGAN-GP with different a sample size  $n \in \{1000, 10000, 100000, 1000000\}$  for the case of  $n_h = 300$ . We also see a very small gap for a large sample size. Fig. 4 shows the performance as a function of iterations when  $n_h = 300$  and the sample size  $n = 1,000,000$ . We can also see a small gap in the convergence.

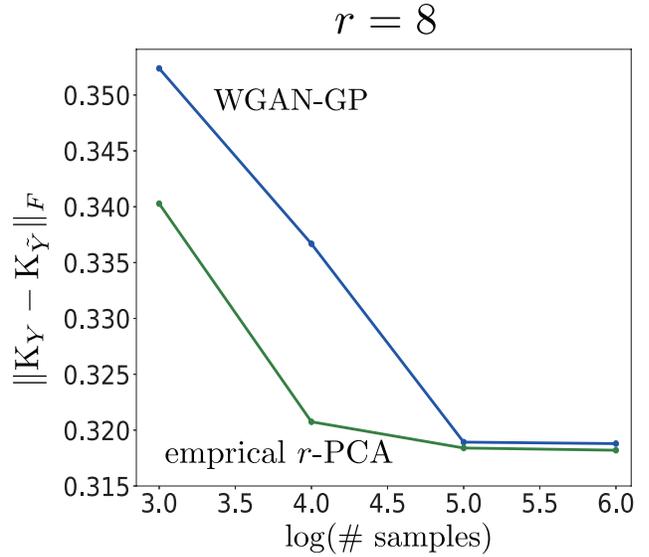


Fig. 5. The comparison between performances of WGAN-GP and the empirical  $r$ -PCA solution while varying a sample size  $n \in \{1000, 10000, 100000, 1000000\}$ . We can observe a non-significant gap to the empirical  $r$ -PCA solution for each sample size and the gap vanishes with an increase in  $n$ .

## VI. DISCUSSION

Our theoretical result focuses on the population limit case. We also explore via some experiments a more practically-relevant setting in which the sample size  $n$  is finite and the optimal ML solution is the *empirical*  $r$ -PCA. Specifically we compare WGAN-GP with the empirical  $r$ -PCA while varying a sample size  $n \in \{1000, 10000, 100000, 1000000\}$ . For the discriminator implementation in WGAN-GP, we employ a 5-layer neural network with 300 hidden units at each layer. See Fig. 5. Observe that the WGAN-GP algorithm converges to the equilibrium with a non-significant gap to the empirical  $r$ -PCA performance, and the gap vanishes with an increase in  $n$ . From simulation results, we conjecture that WGAN optimization may achieve the empirical PCA solution for a finite sample-sized setting. The proof of the conjecture requires verification that the generator design w.r.t. the *empirical* distribution of real data samples yields the empirical PCA solution. If that is not the case (the conjecture is disproved), then there may be a proper constraint for the discriminator that leads to the empirical PCA solution, as in the case of quadratic GAN [13] where the 2nd-order Wasserstein distance together with a *quadratic*-function constraint for the discriminator yields the empirical PCA solution.

## VII. CONCLUSION

We showed that WGAN can recover the optimal PCA solution under the linear-generator Gaussian-data setting. This promises that WGAN may form the basis of an optimal GAN architecture. Our future work includes: (1) Discovering a class of GAN architectures that performs PCA under the Gaussian setting (if any); (2) Exploring other settings in which other

<sup>2</sup>Training of WGAN takes less than 60 seconds on a TESLA\_P40 GPU.

GAN architectures provide optimal solutions (if any); (3) Developing novel algorithms which can resolve stability issues (if required).

#### VIII. ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2018R1A1A1A05022889).

#### REFERENCES

- [1] 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 27*, pp. 2672–2680, 2014.
- [2] S. Nowozin, B. Cseke, and R. Tomioka, "f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization," in *Advances in Neural Information Processing Systems 29*, pp. 271–279, 2016.
- [3] G. K. Dziugaite, D. M. Roy, and Z. Ghahramani, "Training generative neural networks via Maximum Mean Discrepancy optimization," *arXiv preprint arXiv:1505.03906*, 2015.
- [4] Y. Li, K. Swersky, and R. Zemel, "Generative Moment Matching Networks," in *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1718–1727, 2015.
- [5] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," in *Proceedings of the 34th International Conference on Machine Learning*, pp. 214–223, 2017.
- [6] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least Squares Generative Adversarial Networks," in *IEEE International Conference on Computer Vision*, pp. 2813–2821, 2017.
- [7] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: Boundary Equilibrium Generative Adversarial Networks," *arXiv preprint arXiv:1703.10717*, 2015.
- [8] M. E. Tipping and C. M. Bishop, "Probabilistic Principal Component Analysis," *Journal of the Royal Statistical Society, Series B*, vol. 61, no. 3, pp. 611–622, 1999.
- [9] A. Sanjeev, G. Rong, L. Yingyu, M. Tengyu, and Z. Yi, "Generalization and Equilibrium in Generative Adversarial Nets (GANs)," in *Proceedings of the 34th International Conference on Machine Learning*, pp. 224–232, 2017.
- [10] F. Farzan and T. David, "A Convex Duality Framework for GANs," in *Advances in Neural Information Processing Systems 31*, pp. 5248–5258, 2018.
- [11] C. H. Q. Ding, D. Zhou, X. He, and H. Zha, " $R_1$ -PCA: Rotational Invariant  $L_1$ -norm Principal Component Analysis for Robust Subspace Factorization," in *Proceedings of the 23rd International Conference on Machine Learning*, pp. 281–288, 2006.
- [12] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved Training of Wasserstein GANs," in *Advances in Neural Information Processing Systems 30*, pp. 5767–5777, 2017.
- [13] S. Feizi, F. Farnia, T. Ginart, and D. Tse, "Understading GANs: the LQG Setting," *arXiv preprint arXiv:1710.10793*, 2017.
- [14] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York Wiley, 2th ed., July 2006.
- [15] J. Lin, "Divergence Measures Based on the Shannon Entropy," *IEEE Transactions on Information Theory*, vol. 37, pp. 145–151, Jan. 1991.
- [16] C. Villani, *Optimal Transport: Old and New*. New York Wiley, 2th ed., 2009.
- [17] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010.