Uncertainty Principles of Encoding GANs

Ruili Feng [1] **Zhouchen Lin** [‡ 2 3] **Jiapeng Zhu** [4] **Deli Zhao** [5] **Jinren Zhou** [5] **Zheng-Jun Zha** [† 1]

Abstract

The compelling synthesis results of Generative Adversarial Networks (GANs) demonstrate rich semantic knowledge in their latent codes. To obtain this knowledge for downstream applications, encoding GANs has been proposed to learn encoders, such that real world data can be encoded to latent codes, which can be fed to generators to reconstruct those data. However, despite the theoretical guarantees of precise reconstruction in previous works, current algorithms generally reconstruct inputs with non-negligible deviations from inputs. In this paper we study this predicament of encoding GANs, which is indispensable research for the GAN community. We prove three uncertainty principles of encoding GANs in practice: a) the 'perfect' encoder and generator cannot be continuous at the same time, which implies that current framework of encoding GANs is ill-posed and needs rethinking; b) neural networks cannot approximate the underlying encoder and generator precisely at the same time, which explains why we cannot get 'perfect' encoders and generators as promised in previous theories; c) neural networks cannot be stable and accurate at the same time, which demonstrates the difficulty of training and trade-off between fidelity and disentanglement encountered in previous works. Our work may eliminate gaps between previous theories and empirical results, promote the understanding of GANs, and guide network designs for follow-up works.

† Corresponding author, ‡ co-corresponding author. [1]University of Science and Technology of China, Hefei, China. [2]Key Lab. of Machine Perception (MoE), School of EECS, Peking University, Beijing, China. [3]Pazhou Lab, Guangzhou, China. [4]Hong Kong University of Science and Technology, Hong Kong, China. [5]Alibaba Group. Correspondence to: Ruili Feng <frl1996@mail.ustc.edu.cn>, Zhouchen Lin <zlin@pku.edu.cn>, Zheng-Jun Zha <zhazj@ustc.edu.cn>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

1. Introduction

Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) are powerful unsupervised models of establishing maps from simple latent distributions to arbitrarily complex data distributions in various real world scenarios like computer vision (Liang et al., 2017; Zhang et al., 2019; Karras et al., 2019; 2020; Zheng et al., 2020; Liu et al., 2019; Zha et al., 2020), natural language processing (Zhang et al., 2017; Xu et al., 2018; Liu et al., 2018), medicine (Yi & Babyn, 2018; Wolterink et al., 2017; Yi et al., 2018; Frid-Adar et al., 2018), and chemistry (De Cao & Kipf, 2018). Their impressive synthesis performance has aroused a surge of interests in encoding data into latent spaces of GANs for representation learning (Donahue & Simonyan, 2019; Ma et al., 2019; Asim et al., 2020), image editing (Bau et al., 2020; Richardson et al., 2020; Shen & Zhou, 2020; Abdal et al., 2020b), and other downstream tasks (Lin et al., 2019; Rosca et al., 2018; Lewis et al., 2021).

The current framework of encoding GAN researches can be abstracted as follows. Let $\mathcal{Z}$ and $\mathcal{X}$ be the latent space and the data manifold, and $\mathbb{P}_{\mathcal{Z}}$ and $\mathbb{P}_{\mathcal{X}}$ be the latent distribution and the data distribution on $\mathcal{Z}$ and $\mathcal{X}$, respectively. Encoding GANs introduces a bijection between the latent and the data: an underlying 'perfect' generator $\boldsymbol{g}$ transports the latent distribution into the data one,

$$\mathbb{P}_{\boldsymbol{g}(\mathcal{Z})}(\mathcal{A}) = \int_{\boldsymbol{g}^{-1}(\mathcal{A})} d\mathbb{P}_{\mathcal{Z}} = \mathbb{P}_{\mathcal{X}}(\mathcal{A}), \forall \mathcal{A} \subset \mathcal{F}_{\mathcal{X}}, \quad (1)$$

where $\mathcal{F}_{\mathcal{X}}$ is the collection of measurable sets in $\mathcal{X}$; and an underlying 'perfect' encoder $\boldsymbol{e}$ inverts the generator,

$$\boldsymbol{e} \circ \boldsymbol{g}(\boldsymbol{z}) = \boldsymbol{z}, \boldsymbol{g} \circ \boldsymbol{e}(\boldsymbol{x}) = \boldsymbol{x}, \forall \boldsymbol{z} \in \mathcal{Z}, \boldsymbol{x} \in \mathcal{X}. \quad (2)$$

The training algorithms then aim at approximating the underlying 'perfect' encoder and generator with parameterized neural networks $\boldsymbol{E}_{\boldsymbol{\theta}}$ and $\boldsymbol{G}_{\boldsymbol{\phi}}$, respectively.

The above framework of encoding GANs supports both to encode a pre-trained GAN, or to learn a GAN equipped with an encoder in an end-to-end manner, which divides current encoding GAN algorithms into two training methodologies: 1) concurrent training, *i.e.* training the encoder and generator concurrently as in ALI (Dumoulin et al., 2017) and BiGAN (Donahue et al., 2017; Donahue & Simonyan, 2019); 2) two phase training, *i.e.* training an encoder to

invert a fixed and pretrained generator as in (Perarnau et al., 2016; Reed et al., 2016; Zhu et al., 2019). There are theoretical supports for both methodologies. For concurrent training, BiGAN and ALI have proved that neural networks will attain 'perfect' reconstruction and synthesis at the global minimum of training algorithms (Theorem 2 & Proposition 3 in (Donahue et al., 2017; Donahue & Simonyan, 2019)). For two phase training, *Universal Approximation Theorem* (Cybenko, 1989; Pinkus, 1999) says that one layer neural networks can fit a given continuous mapping (the inverse of pretrained generator) arbitrarily well.

Despite the theoretical guarantee that neural networks can approximate the 'perfect' encoder & generator, the practice of encoding GANs is far from satisfactory. Optimization-based GAN inversion methods (Abdal et al., 2019; 2020a; Gabbay & Hoshen, 2019) solve the inverted latent code of data point $\boldsymbol{x}$ by

$$\boldsymbol{z}(\boldsymbol{x}) = \arg\min_{\boldsymbol{z}\in\mathcal{Z}} \|\boldsymbol{G}_{\boldsymbol{\phi}}(\boldsymbol{z}) - \boldsymbol{x}\|^2 + \lambda R(\boldsymbol{z}), \qquad (3)$$

where $R(\boldsymbol{z})$ is the regularization term. They significantly outperform explicit encoders in inversion quality. The encoder and the generator with concurrent training provide informative representations for downstream tasks (Donahue et al., 2017; Dumoulin et al., 2017; Donahue & Simonyan, 2019; Belghazi et al., 2018b; Chen et al., 2016; Belghazi et al., 2018a), but generate less competitive results than state-of-the-art GAN models (Brock et al., 2018; Karras et al., 2019; 2020), and cannot achieve the faithful reconstruction. While two phase training can keep the synthesis quality of generators, it still reconstructs inputs with considerable differences (Perarnau et al., 2016; Reed et al., 2016; Zhu et al., 2019). As both the synthesis and inversion ability are vital for downstream tasks, it is necessary to close the gap between theory and empirical performance, uncover the black box behind encoding GANs, and offer insights to network designs.

Here we provide a theoretical framework for analyzing encoding GANs and handling challenges mentioned above. Different from many theoretical works built on strong assumptions and narrow scenarios like smoothness, Gaussian distributions, or shallow network architectures, we only make three mild assumptions: data lie in a manifold, the neural networks are continuous and piece-wise continuously differentiable, and all involved probability distributions have densities. All assumptions are broadly accepted in the deep learning community (Wold et al., 1987; Candès et al., 2011; LeCun et al., 2015; Goodfellow et al., 2016; Lin et al., 2018), and are consistent with the practice well (Glorot et al., 2011; Ioffe & Szegedy, 2015; Krizhevsky et al., 2017). This allows our theory to be closely connected with the practice, have universal meaning in guiding network designs, and supplement many previous theoretical works in related directions. Our main contributions are summarized as follows:

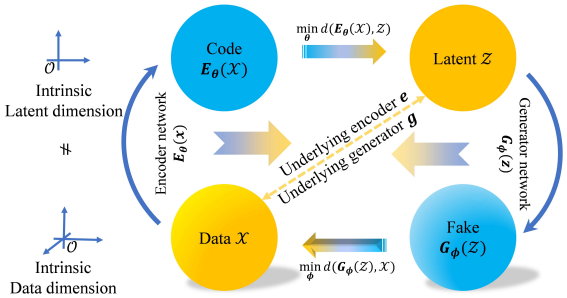


Figure 1. A typical training process. Training algorithms guide encoder network $\boldsymbol{E}_{\boldsymbol{\theta}}$ and generator network $\boldsymbol{G}_{\boldsymbol{\phi}}$ to the underlying $\boldsymbol{e}$ and $\boldsymbol{g}$ by minimizing divergences from $\boldsymbol{E}_{\boldsymbol{\theta}}(\mathcal{X})$ to $\mathcal{Z}$ and from $\boldsymbol{G}_{\boldsymbol{\phi}}(\mathcal{Z})$ to $\mathcal{X}$. It is worthwhile to note that, as the latent distribution is pre-assigned and fixed, we usually have $dim(\mathcal{Z}) \neq dim(\mathcal{X})$.

- Our theory demonstrates three uncertainty principles [1] in the practice of encoding GANs: a) the underlying encoder and generator cannot be continuous at the same time; b) neural networks cannot accurately approximate the underlying encoder and generator at the same time; c) neural networks cannot be stable and accurate at the same time.
- Our theorems explain why we always get 'imperfect' encoders and generators, why we sometimes have unstable training, and why we sometimes encounter trade-off between fidelity and disentanglement (Karras et al., 2019; 2020), despite the theoretical guarantees in (Donahue et al., 2017; Donahue & Simonyan, 2019; Dumoulin et al., 2017; Arjovsky et al., 2017; Arjovsky & Bottou, 2017; Gulrajani et al., 2017). Our theorems also supplement those previous theoretical works.
- We provide examples to validate the three uncertainty principles and provide intuitive understandings on the uncertainty principles.

Although our theoretical analysis is for encoding GANs, we can also apply it to the encoding of other generative models, such as Wassertein auto-encoders (Tolstikhin et al., 2017), adversarial auto-encoders (Makhzani et al., 2015), and auto-encoders with fixed latent distributions.

2. Preliminaries

We start by introducing our settings. See Tab. 1 for meanings and examples of notations used in this paper. Fig. 1 illustrates a typical training process. Training algorithms guide the encoder network $\boldsymbol{E}_{\boldsymbol{\theta}}$ and the generator network $\boldsymbol{G}_{\boldsymbol{\phi}}$ to the underlying 'perfect' encoder $\boldsymbol{e}$ and generator $\boldsymbol{g}$ by

[1] Here we mean that there are always two properties that cannot be reached together, which is similar to the uncertainty principle in physics (Robertson, 1929).

Table 1. Examples of notation used in this paper.

d-dimensional volume	$m_d(\cdot)$
Scalars	$\sigma, \delta, \epsilon, m, n, d$
Scalar value functions	f, g
Vectors	$\boldsymbol{x}, \boldsymbol{z}$
Vector value functions	$\boldsymbol{e}, \boldsymbol{g}$
Sets & Manifolds	$\mathcal{X}, \mathcal{Z}, \mathcal{D}$
Neural networks	$\boldsymbol{E_\theta}, \boldsymbol{G_\phi}, \boldsymbol{D_\psi}$
Distributions	$\mathbb{P}_\mathcal{X}(\boldsymbol{x}), \mathbb{P}_\mathcal{Z}(\boldsymbol{z})$
Induced Distributions	$\mathbb{P}_{\boldsymbol{g}(\mathcal{Z})}(\boldsymbol{x}), \mathbb{P}_{\boldsymbol{E_\theta}(\mathcal{X})}(\boldsymbol{z})$
Intrinsic dimension of manifolds	$dim(\mathcal{X}), dim(\mathcal{Z})$

minimizing divergence from $\boldsymbol{E_\theta}(\mathcal{X})$ to $\mathcal{Z}$ and from $\boldsymbol{G_\phi}(\mathcal{Z})$ to $\mathcal{X}$. Popular divergences include the Jensen-Shannon divergence (Donahue et al., 2017; Dumoulin et al., 2017)

$$D_{JS}(\mathbb{P}_\mathcal{A},\mathbb{Q}_\mathcal{B}) = \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_\mathcal{A}}\left[\log\left(\frac{2\mathbb{P}_\mathcal{A}(d\boldsymbol{x})}{\mathbb{P}_\mathcal{A}(d\boldsymbol{x})+\mathbb{Q}_\mathcal{B}(d\boldsymbol{x})}\right)\right] + \mathbb{E}_{\boldsymbol{y}\sim\mathbb{Q}_\mathcal{B}}\left[\log\left(\frac{2\mathbb{Q}_\mathcal{B}(d\boldsymbol{y})}{\mathbb{P}_\mathcal{A}(d\boldsymbol{y})+\mathbb{Q}_\mathcal{B}(d\boldsymbol{y})}\right)\right], \quad (4)$$

KL divergence (Makhzani et al., 2015), l_2 reconstruction loss (Choi et al., 2020; Li et al., 2017), and Wasserstein divergence (Tolstikhin et al., 2017)

$$W_1(\mathbb{P}_\mathcal{A}, \mathbb{Q}_\mathcal{B}) = \inf_{\pi\in\Pi(\mathbb{P}_\mathcal{A},\mathbb{Q}_\mathcal{B})}\int_{\mathcal{A}\times\mathcal{B}}\|\boldsymbol{x}-\boldsymbol{y}\|\,d\pi(\boldsymbol{x},\boldsymbol{y}), \quad (5)$$

where $\Pi(\mathbb{P}_\mathcal{A}, \mathbb{Q}_\mathcal{B})$ is the collection of all joint distributions of $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{A}\times\mathcal{B}$ which have marginal distribution $\mathbb{P}_\mathcal{A}$ for $\boldsymbol{x}$ and $\mathbb{Q}_\mathcal{B}$ for $\boldsymbol{y}$.

Usually, latent space $\mathcal{Z}$ and data space $\mathcal{X}$ are treated as manifolds embedded in some Euclidean ambient spaces. We introduce the concept of manifolds and their *intrinsic dimensions* (Gallot et al., 1990) below, and give examples in Fig. 2. *Note that throughout this paper, we use the word 'dimension' for the intrinsic dimension of manifold, not the dimension of its ambient space.*

Definition 1 (Intrinsic Dimension and Manifold) *If for any point $\boldsymbol{x} \in \mathcal{A}$, it has a small open neighborhood $\mathcal{U}$ and a continuous bijection $\mathbf{b}$ (also called the **chart** at $\boldsymbol{x}$) that maps $\mathcal{U}\cap\mathcal{A}$ to an open set in $\mathbb{R}^n$, then n is the intrinsic dimension of $\mathcal{A}$. We denote it as $dim(\mathcal{A}) = n$. Accordingly, $\mathcal{A}$ is called a manifold.*

We introduce two specific examples of the training process. The concurrent training process in BiGAN (Donahue et al., 2017) solves a zero-sum game

$$\min_{\boldsymbol{\theta},\boldsymbol{\phi}}\max_{\boldsymbol{\psi}} V(\boldsymbol{E_\theta}, \boldsymbol{G_\phi}, \boldsymbol{D_\psi}), \quad (6)$$

where $\boldsymbol{D_\psi}$ is the discriminator network for the $(\boldsymbol{x}, \boldsymbol{z})$ pair, and

$$V(\boldsymbol{E_\theta}, \boldsymbol{G_\phi}, \boldsymbol{D_\psi}) = \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_\mathcal{X}}\left[\log\left(D(\boldsymbol{x}, \boldsymbol{E_\theta}(\boldsymbol{x}))\right)\right] + \mathbb{E}_{\boldsymbol{z}\sim\mathbb{P}_\mathcal{Z}}\left[1-\log(D(\boldsymbol{G_\phi}(\boldsymbol{z}), \boldsymbol{z}))\right]. \quad (7)$$

Another example is the two phase training process of LIA (Zhu et al., 2019), where a generator is trained by solving

$$\min_{\boldsymbol{\theta}}\max_{\boldsymbol{\psi}} V(\boldsymbol{G_\phi}, \boldsymbol{D_\psi}), \quad (8)$$

in which

$$V(\boldsymbol{G_\phi}, \boldsymbol{D_\psi}) = \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_\mathcal{X}}[\log(D(\boldsymbol{x}))] + \mathbb{E}_{\boldsymbol{z}\sim\mathbb{P}_\mathcal{Z}}[1-\log(D(\boldsymbol{G_\phi}(\boldsymbol{z}))], \quad (9)$$

and then an encoder is trained by optimizing

$$\min_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_\mathcal{X}}[\|\boldsymbol{G_\phi}\circ\boldsymbol{E_\theta}(\boldsymbol{x})-\boldsymbol{x}\|_2^2] + d(\mathbb{P}_{\boldsymbol{E_\theta}(\mathcal{X})}, \mathbb{P}_\mathcal{Z}), \quad (10)$$

in which d is among the divergences of distributions introduced at the beginning of this section.

Current design of generative models assigns a fixed latent distribution to the generator, which also fixes the intrinsic dimension of latent distribution. Specifically, for the popular standard Gaussian latents, the intrinsic dimension is the number of variables (Goodfellow et al., 2014; Gallot et al., 1990). We disallow the networks to adjust the latent distribution during training, because we need each sample $\boldsymbol{z} \in \mathcal{Z}$ from the latent distribution to produce meaningful synthesis in $\mathcal{X}$ through the generator. This is essentially different from auto-encoders (Hinton & Zemel, 1994; Ng et al., 2011) which are not designed for synthesis and allow self-adaptation in the latent distribution. As $dim(\mathcal{X})$ is often unclear, and $dim(\mathcal{Z})$ is manually assigned before training, we are safe to assume that the latent space $\mathcal{Z}$ and domain of interest $\mathcal{X}$ have different intrinsic dimensions, *i.e.* $dim(\mathcal{X}) \neq dim(\mathcal{Z})$.

To build the foundation of our theory, we make the following assumptions, which are almost the minimum requests for theoretical analysis.

Assumption 1 *Throughout this paper, we assume that:*

- *the data domain $\mathcal{X}$ is a manifold with an intrinsic dimension n, where n is unknown;*
- *the neural networks $\boldsymbol{E_\theta}(\boldsymbol{x})$ and $\boldsymbol{G_\phi}(\boldsymbol{z})$ are continuous and piece-wise continuously differentiable with respect to inputs $\boldsymbol{x}$ and $\boldsymbol{z}$; we do not make any assumption on the training method or the loss function;*
- *the latent and the data distributions are absolutely continuous with respect to the Lebesgue measure on $\mathcal{Z}$ and $\mathcal{X}$ respectively, which are the minimum requirements for calculating the Jensen-Shannon and Wasserstein divergences.*

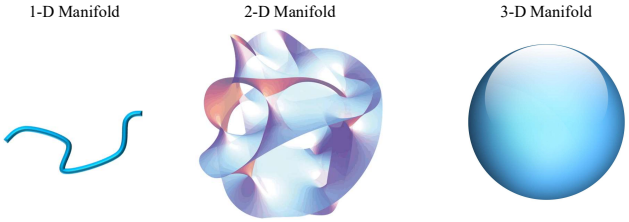


Figure 2. Intrinsic dimensions of manifolds in $\mathbb{R}^3$. All the above sets have 3-D coordinates (x, y, z) in $\mathbb{R}^3$, but their intrinsic dimensions are different.

Remark 1 *Obviously, neural network components such as MLPs, CNNs, Relu, Tanh, LeakyRelu, Softmax, Sigmoid, and neural networks composed of them are all continuous and piece-wise continuously differentiable with respect to their inputs.*

Remark 2 *The readers may note that we do not assume the training technique and architecture details of the generator and encoder. Thus the generator and encoder can also be obtained by other methods like variational inferences (Kingma & Welling, 2013).*

3. Uncertainty Principles

3.1. Uncertainty in the Continuity of the Underlying Encoder and Generator

Our first result suggests that the underlying encoder and generator may not be smooth at the same time.

Theorem 1 *When $dim(\mathcal{Z}) \neq dim(\mathcal{X})$, at least one of the underlying encoder and generator in Eq. (1) & (2) is discontinuous; and for any $\boldsymbol{x} \in \mathcal{X}, \delta > 0$, there is a point $\boldsymbol{x}'$ in the geodesic ball centered at $\boldsymbol{x}$ with radius δ, such that $\boldsymbol{e}$ is not continuous at $\boldsymbol{x}'$ or $\boldsymbol{g}$ is not continuous at $\boldsymbol{e}(\boldsymbol{x}')$. The same thing holds for $\mathcal{Z}$.*

Theorem 1 almost excludes continuous underlying encoders and generators in practice, and underlines that the discontinuous points exist in every neighborhood. It reveals the extremely bad property of the underlying encoder and generator, and nearly excludes the chance for continuous networks to exactly represent the underlying encoder and generator. Also, it urges us to rethink the current design of encoding GANs, as continuity of underlying functions is so important in theoretical foundations of universal approximation abilities of neural networks (Cybenko, 1989; Hornik et al., 1989; Allan, 1999; Hanin & Sellke, 2018; Johnson, 2018; Kidger & Lyons, 2020; Park et al., 2021), representation learning (Bengio et al., 2013), unsupervised learning (Barlow, 1989; Belkin & Niyogi, 2001; Belkin et al., 2006), and other downstream tasks (Belkin & Niyogi, 2003; Zhang & Zha, 2004; Chang et al., 2004). The uncertainty in continuity no doubt strikes the heart of both network designs and downstream tasks of encoding GANs.

3.2. Uncertainty in Universal Approximation Ability

Apart from the above, we are interested in quantitatively analyzing how well the neural networks can approximate the underlying encoder and generator. The *Universal Approximation Theorem* (Pinkus, 1999) states that neural networks can approximate any continuous functions with arbitrary accuracy. However, in this paper, we find that neural networks are seldom universal approximators in encoding GANs.

Theorem 2 *When $dim(\mathcal{Z}) \neq dim(\mathcal{X})$, neural networks are not universal approximators to the underlying encoder and generator in Eq. (1) & (2). More specifically, we have:*

$$\inf_{\boldsymbol{\theta},\boldsymbol{\phi}} \delta_{\boldsymbol{e}}(\boldsymbol{\theta}) + \delta_{\boldsymbol{g}}(\boldsymbol{\phi}) \geq D_{\boldsymbol{e}} + D_{\boldsymbol{g}} > 0, \tag{11}$$

where

$$D_{\boldsymbol{e}} = \frac{1}{2} \sup_{\boldsymbol{x}\in\mathcal{X}} \limsup_{\boldsymbol{y}\to\boldsymbol{x}} \|\boldsymbol{e}(\boldsymbol{y}) - \boldsymbol{e}(\boldsymbol{x})\|, \tag{12}$$

$$D_{\boldsymbol{g}} = \frac{1}{2} \sup_{\boldsymbol{z}\in\mathcal{Z}} \limsup_{\boldsymbol{w}\to\boldsymbol{z}} \|\boldsymbol{g}(\boldsymbol{w}) - \boldsymbol{g}(\boldsymbol{z})\|, \tag{13}$$

and

$$\delta_{\boldsymbol{e}}(\boldsymbol{\theta}) = \sup_{\boldsymbol{x}\in\mathcal{X}} \|\boldsymbol{E}_{\boldsymbol{\theta}}(\boldsymbol{x}) - \boldsymbol{e}(\boldsymbol{x})\|, \tag{14}$$

$$\delta_{\boldsymbol{g}}(\boldsymbol{\phi}) = \sup_{\boldsymbol{z}\in\mathcal{Z}} \|\boldsymbol{G}_{\boldsymbol{\phi}}(\boldsymbol{z}) - \boldsymbol{g}(\boldsymbol{z})\|. \tag{15}$$

Moreover, if $dim(\mathcal{Z}) < dim(\mathcal{X})$, we have

$$D_{JS}(\mathbb{P}_{\boldsymbol{G}_{\boldsymbol{\phi}}(\mathcal{Z})}, \mathbb{P}_{\mathcal{X}}) \geq \frac{\log 2}{2}, \tag{16}$$

and if $dim(\mathcal{X}) < dim(\mathcal{Z})$, we have

$$D_{JS}(\mathbb{P}_{\boldsymbol{E}_{\boldsymbol{\theta}}(\mathcal{X})}, \mathbb{P}_{\mathcal{Z}}) \geq \frac{\log 2}{2}. \tag{17}$$

Theorem 2 seems to contradict *Theorem 2* in BiGAN (Donahue et al., 2017) that the underlying 'perfect' encoder and generator can be reached by the training algorithms. *This contradiction comes from a potential assumption in the proof of BiGAN: the Jensen-Shannon divergence between the induced (encoded or generated) distributions and real (latent or data) distributions can reach exact zero.* This assumption is well consistent with practice when it is originally used in the proof of *Theorem 1* in BiGAN, where $\boldsymbol{E}_{\boldsymbol{\theta}}$ and $\boldsymbol{G}_{\boldsymbol{\phi}}$ have indeterministic architectures, *i.e.* $\boldsymbol{E}_{\boldsymbol{\theta}}(\boldsymbol{x}) = p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta})$ and $\boldsymbol{G}_{\boldsymbol{\phi}}(\boldsymbol{z}) = p(\boldsymbol{x}|\boldsymbol{z};\boldsymbol{\phi})$ are distributions rather than specific vectors given inputs $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{z} \in \mathcal{Z}$. However, in the proof of *Theorem 2* (Appendix

A.3 & A.4 of (Donahue et al., 2017)) in BiGAN, $\boldsymbol{E_\theta}(\boldsymbol{x})$ and $\boldsymbol{G_\phi}(\boldsymbol{z})$ are limited to deterministic functions. This assumption then no longer holds if $dim(\mathcal{Z}) \neq dim(\mathcal{X})$, as we show in Fig. 4 and 3, and Eq. (17) of Theorem 2. It then results in the failure of the theory of BiGAN in practice. The detailed analysis is provided in the supplementary material.

Theorem 2 estimates how close neural networks can approach the underlying encoder and generator. For all neural networks, whatever the depth, width, architectures, and training methods, their approximation error to $\boldsymbol{e}$ and $\boldsymbol{g}$ is larger than $D_{\boldsymbol{e}} + D_{\boldsymbol{g}}$, which is a positive real number when $dim(\mathcal{Z}) \neq dim(\mathcal{X})$ and only depends on the task itself.

We note that the research field of universal approximations also extensively uses another error measure, the l_p distance between neural networks and underlying targets (Lu et al., 2017; Kidger & Lyons, 2020; Park et al., 2021). However, this error measure for approximation may be ill-posed for generative models. The l_p error measure cares more about how a predict (such as predicted class labels) from the input departs from its real value (such as ground-truth labels), while in the scenario of encoding generative models, we care more about how much region of the data or latent distributions are covered by generated distributions or encoded distributions (Eq. (16) & (17) offer such error measure). Those two things are not equivalent. To demonstrate it, consider the continuous target function

$$f_\epsilon(x) = \begin{cases} \frac{\epsilon}{1-\epsilon}x, & 0 \leq x < 1-\epsilon, \\ \frac{1-\epsilon}{\epsilon}x + \frac{2\epsilon-1}{\epsilon}, & 1-\epsilon \leq x \leq 1, \end{cases} \tag{18}$$

where $0 < \epsilon \ll 1$. It is easy to see that $f([0, 1-\epsilon)) = [0, \epsilon), f([1-\epsilon, 1]) = [\epsilon, 1]$. setting $g(x) = \frac{\epsilon}{1-\epsilon}x$, we have $\int_0^1 |f_\epsilon - g| dx < \epsilon \approx 0$, and $m_1(f_\epsilon([0,1]) \setminus g([0,1])) = 1 - \frac{\epsilon}{1-\epsilon} \approx 1$, where m_1 is the Lebesgue measure on $[0, 1]$. As ϵ is very small, we have that g approximates f_ϵ very well in l_1 error, but most of the output of f_ϵ is not covered by g. Thus, for encoding generative models, the uniform approximation error in Eq. (11) and distribution divergence in Eq. (16) & (17) are more meaningful.

Theorem 2 explains the gap between practice and theory of encoding GANs we have mentioned in Introduction. For optimization-based GAN inversion methods, they do not need a continuous explicit encoder (Creswell & Bharath, 2018; Abdal et al., 2019; 2020a), thus do not yield the error bounds in Eq. (11) & (17). As a consequence, the optimization-based methods may approximate the inverse mapping more accurately than encoder-based methods if provided suitable initialization (Zhu et al., 2019). For explicit encoders and generators (Perarnau et al., 2016; Donahue et al., 2017; Rosca et al., 2017; Su, 2019; Donahue & Simonyan, 2019; Zhu et al., 2019; Pidhorskyi et al., 2020), the joint approximation errors in Eq. (11) & (17) & (16) deviate at least one of the encoder and generator from high quality outputs. In this case, neural networks are not universal approximators for the underlying encoder and generator, and the universal approximation theorem (Pinkus, 1999) does not hold in our scenario.

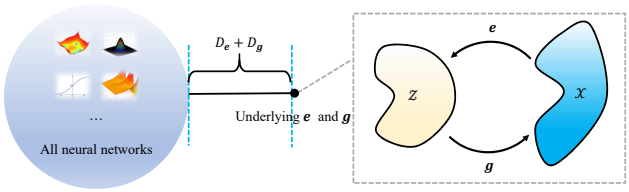


Figure 3. Illustration of Eq. (11). For all the neural networks, no matter their architectures and parameters, they admit positive distance to the underlying encoder and generator, as long as the conditions of Theorem 2 hold.

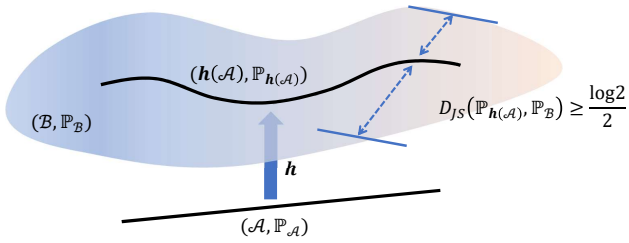


Figure 4. Illustration of Eq. (17). Here we give a simple example. If the latent distribution $\mathbb{P}_\mathcal{A}$ is supported on a real line $\mathcal{A}$, a one-dimensional manifold, and the data distribution $\mathbb{P}_\mathcal{B}$ is supported on a two-dimensional manifold $\mathcal{B}$. Then for any differentiable function $\boldsymbol{h} : \mathcal{A} \to \mathcal{B}$, $\boldsymbol{h}(\mathcal{A})$ is a curve on $\mathcal{B}$. The curve can never occupy the whole 2-D surface $\mathcal{B}$, and thus the Jensen-Shannon divergence can never reach exact zero.

3.3. Uncertainty in Training Dynamics

Our last result digs into the training dynamics. It finds that in most cases gradient explosion cannot be avoided during training, and offers an estimation on the explosion speed.

Theorem 3 *Denote $n = dim(\mathcal{X})$ and $d = dim(\mathcal{Z})$. Let $m_d(\mathcal{Z})$ and $m_n(\mathcal{X})$ be the volumes of $\mathcal{Z}$ and $\mathcal{X}$ with respect to their intrinsic dimensions, respectively. Assume that $\mathcal{Z}$ and $\mathcal{X}$ are bounded manifolds embedded in high dimensional Euclidean spaces, but are almost everywhere diffeomorphism to open subsets in $\mathbb{R}^d$ and $\mathbb{R}^n$, respectively. Denote $diam(\mathcal{Z}) = \sup_{\boldsymbol{z},\boldsymbol{w}\in\mathcal{Z}} \|\boldsymbol{z} - \boldsymbol{w}\|, diam(\mathcal{X}) = \sup_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{X}} \|\boldsymbol{x} - \boldsymbol{y}\|$, and ω_i to be the volume of unit ball of dimension i. For simplicity, let $i, j \in \{d, n\}$ and*

$$\begin{aligned} &\Gamma(\mathcal{A}, \mathcal{B}, i, j, a, b) \\ &= \left(\frac{diam(\mathcal{A})^{j-i} m_j(\mathcal{B})}{3a(2^j m_j(\mathcal{B}) + \omega_j diam(\mathcal{A})^j)} \right)^{\frac{1}{j-i}} \frac{b m_j(\mathcal{B})}{3}. \end{aligned} \tag{19}$$

Then there is a trade-off between the approximation error and the maximum gradient norm of networks if $dim(\mathcal{Z}) \neq$

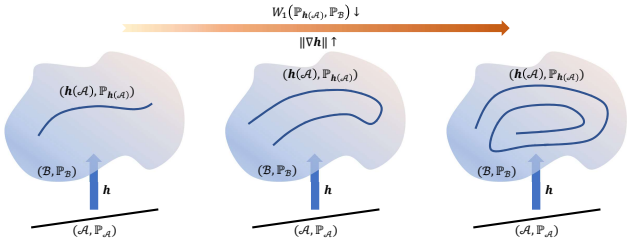


Figure 5. Illustration of Theorem 3. For a one-dimensional curve, the only way to fit a two-dimensional manifold is to twist itself, so that it can occupy more areas.

$dim(\mathcal{X})$. *Specifically, if* $dim(\mathcal{Z}) < dim(\mathcal{X})$, *there exist constants* $C_{\mathcal{X}} > 0$ *that only depends on* $\mathbb{P}_{\mathcal{X}}$ *and* $C_d > 0$ *that only depends on* d, *such that*

$$\begin{aligned} &W_1(\mathbb{P}_{\boldsymbol{G}_{\boldsymbol{\phi}}(\mathcal{Z})}, \mathbb{P}_{\mathcal{X}}) \left(\sup_{\boldsymbol{z}\in\mathcal{Z}} \|\nabla \boldsymbol{G}_{\boldsymbol{\phi}}\| + 1\right)^{\frac{n}{n-d}} \\ \geq &\Gamma(\mathcal{Z}, \mathcal{X}, d, n, C_d, C_{\mathcal{X}}); \end{aligned} \tag{20}$$

if $D_{JS}(\mathbb{P}_{\boldsymbol{G}_{\boldsymbol{\phi}}(\mathcal{Z})}, \mathbb{P}_{\mathcal{X}})) < \log 2$, *then we further have*

$$\begin{aligned} &\frac{D_{JS}(\mathbb{P}_{\boldsymbol{G}_{\boldsymbol{\phi}}(\mathcal{Z})}, \mathbb{P}_{\mathcal{X}})(\sup_{\boldsymbol{z}\in\mathcal{Z}} \|\nabla \boldsymbol{G}_{\boldsymbol{\phi}}\| + 1)^{\frac{2n}{n-d}}}{(diam(\mathcal{Z})(\sup_{\boldsymbol{z}\in\mathcal{Z}} \|\nabla \boldsymbol{G}_{\boldsymbol{\phi}}\|) + diam(\mathcal{X}))^2} \\ \geq &4\Gamma(\mathcal{Z}, \mathcal{X}, d, n, C_d, C_{\mathcal{X}})^2. \end{aligned} \tag{21}$$

On the other hand, if $dim(\mathcal{Z}) > dim(\mathcal{X})$, *there exist constants* $C_{\mathcal{Z}} > 0$ *that only depends on* $\mathbb{P}_{\mathcal{Z}}$ *and* $C_n > 0$ *that only depends on* n, *such that*

$$\begin{aligned} &W_1(\mathbb{P}_{\boldsymbol{E}_{\boldsymbol{\theta}}(\mathcal{X})}, \mathbb{P}_{\mathcal{Z}}) \left(\sup_{\boldsymbol{x}\in\mathcal{X}} \|\nabla \boldsymbol{E}_{\boldsymbol{\theta}}\| + 1\right)^{\frac{n}{d-n}} \\ \geq &\Gamma(\mathcal{X}, \mathcal{Z}, n, d, C_n, C_{\mathcal{Z}}); \end{aligned} \tag{22}$$

if $D_{JS}(\mathbb{P}_{\boldsymbol{E}_{\boldsymbol{\theta}}(\mathcal{X})}, \mathbb{P}_{\mathcal{Z}})) < \log 2$, *then we further have*

$$\begin{aligned} &\frac{D_{JS}(\mathbb{P}_{\boldsymbol{E}_{\boldsymbol{\theta}}(\mathcal{X})}, \mathbb{P}_{\mathcal{Z}})(\sup_{\boldsymbol{x}\in\mathcal{X}} \|\nabla \boldsymbol{E}_{\boldsymbol{\theta}}\| + 1)^{\frac{2n}{d-n}}}{(diam(\mathcal{X})(\sup_{\boldsymbol{x}\in\mathcal{X}} \|\nabla \boldsymbol{E}_{\boldsymbol{\theta}}\|) + diam(\mathcal{Z}))^2} \\ \geq &4\Gamma(\mathcal{X}, \mathcal{Z}, n, d, C_n, C_{\mathcal{Z}})^2, \end{aligned} \tag{23}$$

where W_1 *is the 1-Wasserstein distance (Villani, 2008).*

Remark 3 *Typical examples of manifolds satisfying conditions of Theorem 3 are spheres, hyperbolic surfaces, ellipsoids and their deformed shapes embedded in high dimensional spaces.*

Remark 4 *As we do not impose conditions on the training process, we can apply Theorem 3 to the training of GANs, which is the first phase learning of the two phase encoding GANs.*

Remark 5 *Note that the value of* Γ *is from two positive constants when* $\mathbb{P}_{\mathcal{Z}}$ *and* $\mathbb{P}_{\mathcal{X}}$ *are given.* $\Gamma^{\frac{j-i}{j}}$ *(the explosion speed of maximum gradient norm) grows larger for fixed* j *when* i *decreases or* $m_j(\mathcal{B})$ *increases. It is very small when* $diam(\mathcal{A})$ *is very large. Those changes of values are consistent with our intuition in Fig. 5. However, in order to get a uniform format holding for all situations, we use very loose estimation when* $diam(\mathcal{A})$ *is tiny in the deduction. This makes the value of* Γ *not optimal when* $diam(\mathcal{A})$ *is extremely small and we can use better estimation for this case. See the proof in the supplementary material for details.*

Remark 6 *Note that the Jensen-Shannon divergence has a universal upper bound* $\log 2$. *Thus the condition for Eq. (21) & (23) to hold is equivalent to saying that the induced distribution of encoder or generator network is not too far away from the real distribution.*

Theorem 3 reveals a trade-off between the Wasserstein distance, which is often the training loss in practice, and the maximum gradient norm of networks, if $dim(\mathcal{Z}) \neq dim(\mathcal{X})$. The theorem can be understood intuitively in the following way: the only way for a one-dimensional curve to fit a two-dimensional surface, is to twist the curve so that it can occupy as many areas as possible. We illustrate this intuition in Fig. 5. It means that the training of encoding GANs can be rather unstable and difficult, if both $W_1(\mathbb{P}_{\boldsymbol{e}(\mathcal{X})}, \mathbb{P}_{\mathcal{Z}})$ and $W_1(\mathbb{P}_{\boldsymbol{g}(\mathcal{Z})}, \mathbb{P}_{\mathcal{X}})$ are minimized. Some previous works (Bengio et al., 2013; Belkin & Niyogi, 2003) on representation learning argue that a gentle gradient norm is necessary for good representations computed by networks. Thus Theorem 3 may also suggest bad representation quality when the Wasserstein distance is small.

In a more practical setting, the data distribution is an empirical approximation to the real underlying distribution. For Wasserstein distance, we then have:

Corollary 1 *Under the condition of Theorem 3, and let* $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$ *be* N *independent samples from* $\mathbb{P}_{\mathcal{X}}$. *Let* $\mathbb{Q}_N = \frac{1}{N}\sum_{i=1}^N \delta_{\boldsymbol{x}_i}$ *be the empirical distribution of those samples, where* $\delta_{\boldsymbol{x}}$ *is the Dirac distribution for sample* $\boldsymbol{x}$. *Further assume that* $\int_{\mathcal{X}} \|\boldsymbol{x} - \boldsymbol{y}\|\mathbb{P}_{\mathcal{X}}(d\boldsymbol{y}) < \infty, \forall \boldsymbol{x} \in \mathcal{X}$. *If* $d < n$, *then there exists a constant* $C > 0$ *such that for all generator network* $\boldsymbol{G}_{\boldsymbol{\phi}}$

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{x}_1,...,\boldsymbol{x}_N\sim\mathbb{P}_{\mathcal{X}}}[W_1(\mathbb{P}_{\boldsymbol{G}_{\boldsymbol{\phi}}(\mathcal{Z})}, \mathbb{Q}_N)] \\ \geq &\frac{C}{(\sup_{\boldsymbol{z}\in\mathcal{Z}} \|\nabla \boldsymbol{G}_{\boldsymbol{\phi}}\| + 1)^{\frac{n}{n-d}}} - \mathcal{O}(N^{-\frac{1}{n}}). \end{aligned} \tag{24}$$

Theorem 5 and Corollary 1 point out a case where Wasserstein GANs (Arjovsky et al., 2017; Arjovsky & Bottou, 2017) may suffer from gradient explosion. Wasserstein GANs are proposed to cope with the gradient explosion issue in GANs. They replace the original Jensen-Shannon

divergence (Goodfellow et al., 2014) of GANs with the Wasserstein distance. Wasserstein distance is proved to be more stable for training than the Jensen-Shannon divergence as it is smoother (Arjovsky & Bottou, 2017). But previous works did not discuss whether Wasserstein divergence can totally exclude gradient explosion. A recent theoretical analysis (Bottou et al., 2019) points out that the Monge-Ampère formulation (Villani, 2003; 2008) of WGAN may not have good duality. In this paper, we further give a negative answer in Eq. (20). When $dim(\mathcal{Z}) < dim(\mathcal{X})$, $W_1(\mathbb{P}_{\boldsymbol{G}_{\boldsymbol{\phi}}(\mathcal{Z})}, \mathbb{P}_{\mathcal{X}}) \to 0$ implies $\sup_{\boldsymbol{z}\in\mathcal{Z}} \|\nabla \boldsymbol{G}_{\boldsymbol{\phi}}\| \to \infty$. If the training process meets the exploding points of $\nabla \boldsymbol{G}_{\boldsymbol{\phi}}$, the network will then have gradient explosion.

Theorem 3 also reveals the trade-off between fidelity and disentanglement of GANs when $dim(\mathcal{Z}) < dim(\mathcal{X})$. Specifically, there is a trade-off between the Fréchect Inception Distance (FID) (Heusel et al., 2017) and Path Perceptual Length (PPL) (Karras et al., 2019). PPL is introduced in StyleGAN (Karras et al., 2019; 2020) to measure the semantic disentanglement of generators as:

$$l_p = \mathbb{E}_{\boldsymbol{z}_1,\boldsymbol{z}_2\sim\mathbb{P}_{\mathcal{Z}}}\Big[\frac{1}{\epsilon^2}\|\boldsymbol{V}_{\boldsymbol{\beta}}\circ\boldsymbol{G}_{\boldsymbol{\phi}}(t\boldsymbol{z}_1+(1-t)\boldsymbol{z}_2) - \boldsymbol{V}_{\boldsymbol{\beta}}\circ\boldsymbol{G}_{\boldsymbol{\phi}}((t+\epsilon)\boldsymbol{z}_1+(1-t-\epsilon)\boldsymbol{z}_2)\|_2^2\Big], \tag{25}$$

where $t \in (0,1)$ and $0 < \epsilon \ll 1$ are constants and $\boldsymbol{V}_{\boldsymbol{\beta}}$ is the pretrained VGG network. By the chain rule of differentiation, it is easy to see that $\sup_{\boldsymbol{z}\in\mathcal{Z}} \|\nabla \boldsymbol{G}_{\boldsymbol{\phi}}\|$ has positive correlation with PPL score if the computation of expectation meets $\boldsymbol{z}^* = \arg\sup_{\boldsymbol{z}\in\mathcal{Z}} \|\nabla \boldsymbol{G}_{\boldsymbol{\phi}}\|$ or a sequence of $\boldsymbol{z}_k$ converging to it. As the FID is smaller when the generated and data distributions get closer, a lower FID suggests lower Wasserstein distance, which by Eq. (20) suggests higher maximum gradient norms, and results in a higher PPL score (see *Fig. 9* of (Karras et al., 2019) for the opposite trend of FID and PPL in training).

4. Validating the Uncertainty Principles

This section presents a toy example to illustrate and support our theory. The toy example aims to learn the underlying encoder and generator between uniform distributions of supports of intrinsic dimensions in 1 or 2. The encoder, generator, and discriminator networks consist of 3-layer MLPs with LeakyRelu activations. The numbers of hidden units are 10, 100, and 10 for each MLP layer, to model upsampling and downsampling in typical generative models like StyleGANs (Karras et al., 2019; 2020). Considering the simplicity of the task, we think such shallow architectures are adequate for our purpose.

We train the networks with both concurrent training and two phase training methods. For concurrent training, we use the objective (6) as in BiGAN (Donahue et al., 2017);

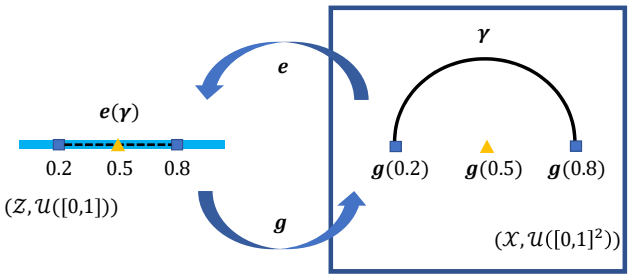


Figure 6. Illustration of the impossibility of continuous underlying encoder and generator between a real line segment in $\mathbb{R}$ and a unit square in $\mathbb{R}^2$. Refer to Section 4.1 for detail.

for two phase training, we use the objectives (8) and (10) with $d = d_{\mathcal{Z}} + d_{recon}$, where $d_{\mathcal{Z}}$ is the Jensen-Shannon divergence between the encoder output and the latent space, and d_{recon} is the Jensen-Shannon divergence between the reconstructed data distribution and the real data distribution.

For the zero-sum game in objectives (6), (8), (10), we solve it by the adversarial training process in *Algorithm 1* of (Goodfellow et al., 2014). For each experiment setting, we further change the number of steps to apply to the discriminator (see *Algorithm 1* of (Goodfellow et al., 2014) for meaning of it) in each adversarial training step. For two phase training, as there are multiple discriminators, we take the following strategy: when $dim(\mathcal{Z}) < dim(\mathcal{X})$, we change the steps of discriminators which discriminate the generators' outputs and real data; when $dim(\mathcal{Z}) > dim(\mathcal{X})$, we change the steps of discriminators which discriminate the encoders' outputs and the latents. More discriminator steps produce more discriminative discriminators, thus the corresponding generators or encoders have to align their outputs to real distributions more precisely. We are going to explore how this influences the results.

4.1. Uncertainty in the Continuity of the Underlying Encoder and Generator

Theorem 1 is difficult to verify experimentally, as we do not know the exact form of the underlying encoder & generator. However, we can infer the property of the underlying encoder & generator from the basic geometric result:

Lemma 1 *If $\boldsymbol{h}$ is continuous and $\mathcal{D}$ is connected, then $\boldsymbol{h}(\mathcal{D})$ is also connected (Rudin et al., 1964).*

Let $\mathcal{U}([0,1])$ be the latent $\mathbb{P}_{\mathcal{Z}}$ and $\mathcal{U}([0,1]^2)$ be the data distribution $\mathbb{P}_{\mathcal{X}}$. Assume that both the underlying generator $\boldsymbol{g}$ and encoder $\boldsymbol{e}$ are continuous. Then it is easy to see that $\boldsymbol{g}$ maps $[0,1]\setminus\{0.5\}$ to $[0,1]^2\setminus\{\boldsymbol{g}(0.5)\}$, and $\boldsymbol{e}$ performs the inversion. This is, however, against Lemma 1, as $[0,1]\setminus\{0.5\}$ is not connected, while $[0,1]^2\setminus\{\boldsymbol{g}(0.5)\}$ is obviously connected, as shown in Fig. 6.

4.2. Uncertainty in Universal Approximation Ability

We now check whether our toy networks can approximate the underlying encoder & generator. We can evaluate it with divergence between induced distributions (by encoder or generator network) and real distributions (of latents or data). The results are reported in Fig. 8 & 7.

It may be no surprise to see that the induced distribution is a curve while the real distribution is a surface area for encoders and generators that try to transfer $\mathcal{U}([0,1])$ to $\mathcal{U}([0,1]^2)$, regardless of the training algorithms. As curves do not hold positive surface area, this means that the induced distributions totally fail to capture the real distributions and the divergence between them should be considerable.

The above observation, however, is a little weak because: 1) our network design or training method may not be optimal; 2) we are not able to exactly estimate the error bound $D_{\boldsymbol{e}} + D_{\boldsymbol{g}}$ to support Eq. (11) of Theorem 2 directly. Fortunately, we can check Eq. (17) of Theorem 2, which is the key to the failure of theories in BiGAN (Donahue et al., 2017) and ALI (Dumoulin et al., 2017), by the following lemma of (Goodfellow et al., 2014):

Lemma 2 (Estimation of the JS Divergence) *For a fixed generator $\boldsymbol{G}$, when the discriminator $\boldsymbol{D}$ is optimal, we have*

$$D_{JS}(\mathbb{P}_{\boldsymbol{G}}, \mathbb{P}_{data}) = \log 2 + \frac{1}{2} V(\boldsymbol{D}, \boldsymbol{G}), \tag{26}$$

where

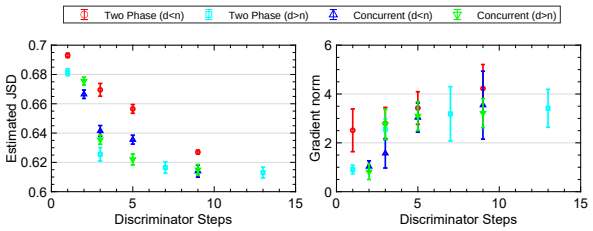
$$\begin{aligned} V(\boldsymbol{D}, \boldsymbol{G}) = \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_{\mathcal{X}}}[\log(\boldsymbol{D}(\boldsymbol{x}))] \\ + \mathbb{E}_{\boldsymbol{z}\sim\mathbb{P}_{\mathcal{Z}}}[\log(1 - \boldsymbol{D}(\boldsymbol{G}(\boldsymbol{z})))]. \end{aligned} \tag{27}$$

By Lemma 2, we develop the following strategy to estimate the Jensen-Shannon divergence between the generated distribution and the data distribution. For a given generator $\boldsymbol{G}$, we fix it and maximize $V(\boldsymbol{D}, \boldsymbol{G})$ until convergence. By Lemma 2, the maximum value of $\frac{1}{2}V(\boldsymbol{D}, \boldsymbol{G})$ plus $\log 2$ offers an estimation to the Jensen-Shannon divergence.

The results are reported in Fig. 7, where we find that the error bound $\frac{\log 2}{2}$ for the Jensen-Shannon divergence always holds. We then look into a different case that the latent is two-dimensional standard Gaussian $\mathcal{N}_2(\mathbf{0}, \mathbf{1})$ and the data distribution is still $\mathcal{U}([0,1]^2)$, to see what would happen if $dim(\mathcal{Z}) = dim(\mathcal{X})$. The networks then show an estimated Jensen-Shannon divergence smaller than $\frac{\log 2}{2}$. The experiments thus can verify that $dim(\mathcal{Z}) \neq dim(\mathcal{X})$ really forces a positive lower bound on the Jensen-Shannon divergence, which does not appear if $dim(\mathcal{Z}) = dim(\mathcal{X})$, as claimed in Theorem 2.

4.3. Uncertainty in Training Dynamics

Theorem 3 claims an increasing maximum norm of the gradient when the approximation gets more accurate. We are curious about how well the practice yields to this theorem.



(a) Estimated JSD and gradient norm ($d \neq n$)

Methods	Encoded	Generated
Concurrent	0.108 ($\pm$ 0.0062)	0.0418 ($\pm$ 0.0045)
Two-phase	0.118 ($\pm$ 0.0064)	0.110 ($\pm$ 0.0070)
Bound in (17)	$\frac{\log 2}{2} \approx 0.347$	

(b) Estimated JSD ($d = n$)

Figure 7. Estimated Jensen-Shannon divergence and gradient norms of networks for distributions in Fig. 8. When $d = dim(\mathcal{Z}) \neq n = dim(\mathcal{X})$, the estimated JSD is always larger than $\frac{\log 2}{2} \approx 0.347$, and the gradient norm increases as there are more steps in training discriminators. When $dim(\mathcal{Z}) = dim(\mathcal{X}) = 2$, the estimated JSD is smaller than $\frac{\log 2}{2}$.

We find the following interesting facts in Fig. 7 & 8: 1) increasing the number of steps to train discriminators makes the generated or encoded distribution 'longer'; 2) the generated and encoded distributions are twisted curves in $\mathbb{R}^2$, and we can increase the cycles of twisting by increasing the number of steps to apply to discriminators; 3) the norms of gradients of the generator or encoder network do increase as D_{JS} gets smaller.

Recall the intuition that inspires Theorem 3, that the only way for a one-dimensional curve to fit a two-dimensional surface, is to twist the curve so that it can occupy as many areas as possible. Experimental results in Fig. 8 support this intuition (Fig. 5) and Theorem 3 behind it, which could be the reason to the surprisingly more and more twisted structure when adding more steps to train discriminators.

As the cycle of the twist grows, the length of the curve of generated distribution also increases. This suggests an increasing gradient norm, as the length of the curve is calculated from the integral on its gradient norm

$$Length(\boldsymbol{\gamma}) = \int_0^1 \|\boldsymbol{\gamma}'(t)\|_2 dt. \tag{28}$$

This observation can be generalized to the following lemma:

Lemma 3 *For differentiable map $\boldsymbol{h} : \mathcal{D} \to \mathbb{R}^d$, $\mathcal{D} \subset \mathbb{R}^d$,*

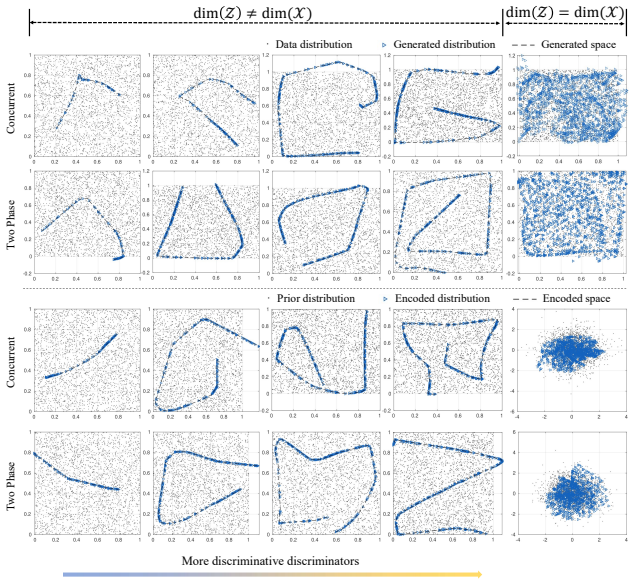


Figure 8. Induced distribution and real distribution of toy examples. Odd rows are results of concurrent training, and even rows are results of two phase training. Except the last column, the first two rows report the encoder outputs when $dim(\mathcal{Z}) = 2, dim(\mathcal{X}) = 1$ and the last two rows report the generator outputs when $dim(\mathcal{Z}) = 1$ and $dim(\mathcal{X}) = 2$. The last column reports the encoder and generator outputs when $dim(\mathcal{Z}) = dim(\mathcal{X}) = 2$ to provide a comparison with the cases of unequal dimensions. We can see that increasing the discriminative ability of discriminators forces the induced distribution of encoders and generators to be more twisted when intrinsic dimensions of latent and data spaces are unequal.

we have

$$m_d(\boldsymbol{h}(\mathcal{D})) \leq \left(\sup_{\boldsymbol{x}\in\mathcal{D}} \|\nabla \boldsymbol{h}\|\right)^d m_d(\mathcal{D}), \tag{29}$$

where $m_d(\cdot)$ is the volume of set in $\mathbb{R}^d$.

In high dimensional settings, Lemma 3 suggests that the volume of the output space is connected with the maximum norm of gradient. We can infer that, for a high dimensional manifold, the only way to fit a manifold of even higher dimension is still by twisting. While twisting means larger volume, Lemma 3 suggests gradient explosion in this case, regardless of training methods and losses. This supports Theorem 3 and generalizes it to broader cases.

5. Conclusion

In this paper, we investigate why encoding GANs are so difficult to achieve their theoretical performance in previous works (Donahue et al., 2017; Dumoulin et al., 2017; Pinkus, 1999). We find that three uncertainty principles deviate the practice from those previous theoretical works. The uncovered uncertainty principles give a quantifiable description to the defects of current frameworks, explain the previous empirical findings of the difficulties (Donahue et al., 2017; Donahue & Simonyan, 2019; Zhu et al., 2019), and reveal fundamental factors in the black box of encoding GANs, such as smoothness, approximation ability, and fitting stability. For each uncertainty principle, we provide simple geometric intuition to demonstrate it. Our theories will serve as a solid starting point of further understanding of encoding GANs and other generative models.

Acknowledgement

This work is supported by the National Key R&D Program of China under Grant 2020AAA0105702, National Natural Science Foundation of China (NSFC) under Grants U19B2038, and the University Synergy Innovation Program of Anhui Province under Grants GXXT-2019-025. Z. Lin is supported by the National Natural Science Foundation of China (Grant No.s 61625301 and 61731018), Project 2020BD006 supported by PKU-Baidu Fund, Major Scientific Research Project of Zhejiang Lab (Grant No.s 2019KB0AC01 and 2019KB0AB02), and Beijing Academy of Artificial Intelligence. Finally, the authors sincerely express vehement protestations of gratitude to Liao Wang of Rheinische Friedrich-Wilhelm Universität Bonn for his help and support.

References

Abdal, R., Qin, Y., and Wonka, P. Image2StyleGAN: How to embed images into the StyleGAN latent space? In *Proceedings of the IEEE international conference on computer vision*, pp. 4432–4441, 2019.

Abdal, R., Qin, Y., and Wonka, P. Image2StyleGAN++: How to edit the embedded images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8296–8305, 2020a.

Abdal, R., Zhu, P., Mitra, N., and Wonka, P. StyleFlow: Attribute-conditioned exploration of StyleGAN-generated images using conditional continuous normalizing flows. *arXiv e-prints*, pp. arXiv–2008, 2020b.

Allan, P. Approximation theory of the MLP model in neural networks [j]. *Acta Numerica*, 8:143–195, 1999.

Arjovsky, M. and Bottou, L. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017.

Asim, M., Daniels, M., Leong, O., Ahmed, A., and Hand, P. Invertible generative models for inverse problems:

Mitigating representation error and dataset bias. In *International Conference on Machine Learning*, pp. 399–409. PMLR, 2020.

Barlow, H. B. Unsupervised learning. *Neural computation*, 1(3):295–311, 1989.

Bau, D., Strobelt, H., Peebles, W., Zhou, B., Zhu, J.-Y., Torralba, A., et al. Semantic photo manipulation with a generative image prior. *arXiv preprint arXiv:2005.07727*, 2020.

Belghazi, M. I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, D. Mutual information neural estimation. In *International Conference on Machine Learning*, pp. 531–540. PMLR, 2018a.

Belghazi, M. I., Rajeswar, S., Mastropietro, O., Rostamzadeh, N., Mitrovic, J., and Courville, A. Hierarchical adversarially learned inference. *arXiv preprint arXiv:1802.01071*, 2018b.

Belkin, M. and Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Nips*, pp. 585–591, 2001.

Belkin, M. and Niyogi, P. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

Belkin, M., Niyogi, P., and Sindhwani, V. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(11), 2006.

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1798–1828, 2013.

Bottou, L., Arjovsky, M., Lopez-Paz, D., and Oquab, M. Geometrical insights for implicit generative modeling, 2019.

Brock, A., Donahue, J., and Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Candès, E. J., Li, X., Ma, Y., and Wright, J. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3): 1–37, 2011.

Chang, H., Yeung, D.-Y., and Xiong, Y. Super-resolution through neighbor embedding. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pp. I–I. IEEE, 2004.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. *arXiv preprint arXiv:1606.03657*, 2016.

Choi, K., Hawthorne, C., Simon, I., Dinculescu, M., and Engel, J. Encoding musical style with transformer autoencoders. In *International Conference on Machine Learning*, pp. 1899–1908. PMLR, 2020.

Creswell, A. and Bharath, A. A. Inverting the generator of a generative adversarial network (II), 2018.

Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

De Cao, N. and Kipf, T. MolGAN: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*, 2018.

Donahue, J. and Simonyan, K. Large scale adversarial representation learning. In *Advances in Neural Information Processing Systems*, pp. 10542–10552, 2019.

Donahue, J., Krähenbühl, P., and Darrell, T. Adversarial Feature Learning, 2017.

Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M., and Courville, A. Adversarially Learned Inference, 2017.

Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., and Greenspan, H. GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing*, 321:321–331, 2018.

Gabbay, A. and Hoshen, Y. Style generator inversion for image enhancement and animation. *arXiv preprint arXiv:1906.11880*, 2019.

Gallot, S., Hulin, D., and Lafontaine, J. *Riemannian geometry*, volume 2. Springer, 1990.

Glorot, X., Bordes, A., and Bengio, Y. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323, 2011.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. *Deep Learning*. MIT press Cambridge, 2016.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks, 2014.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of Wasserstein GANs. In *Advances in neural information processing systems*, pp. 5767–5777, 2017.

Hanin, B. and Sellke, M. Approximating continuous functions by ReLU nets of minimal width. *URL http://arxiv.org/abs/1710.11278*, 2018.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. 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, 2017.

Hinton, G. E. and Zemel, R. S. Autoencoders, minimum description length and Helmholtz free energy. In *Advances in neural information processing systems*, pp. 3–10, 1994.

Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

Johnson, J. Deep, skinny neural networks are not universal approximators. *arXiv preprint arXiv:1810.00393*, 2018.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and improving the image quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8110–8119, 2020.

Kidger, P. and Lyons, T. Universal approximation with deep narrow networks. In *Conference on Learning Theory*, pp. 2306–2327. PMLR, 2020.

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

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521(7553):436–444, 2015.

Lewis, K. M., Varadharajan, S., and Kemelmacher-Shlizerman, I. VOGUE: Try-On by StyleGAN interpolation optimization. *arXiv preprint arXiv:2101.02285*, 2021.

Li, C., Liu, H., Chen, C., Pu, Y., Chen, L., Henao, R., and Carin, L. Alice: Towards understanding adversarial learning for joint distribution matching. *arXiv preprint arXiv:1709.01215*, 2017.

Liang, X., Hu, Z., Zhang, H., Gan, C., and Xing, E. P. Recurrent topic-transition GAN for visual paragraph generation. In *Proceedings of the IEEE international conference on computer vision*, pp. 3362–3371, 2017.

Lin, C. H., Chang, C.-C., Chen, Y.-S., Juan, D.-C., Wei, W., and Chen, H.-T. COCO-GAN: Generation by parts via conditional coordinating. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4512–4521, 2019.

Lin, Z., Khetan, A., Fanti, G., and Oh, S. PacGAN: The power of two samples in generative adversarial networks. *Advances in neural information processing systems*, 2018.

Liu, J., Zha, Z.-J., Chen, D., Hong, R., and Wang, M. Adaptive transfer network for cross-domain person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7202–7211, 2019.

Liu, L., Lu, Y., Yang, M., Qu, Q., Zhu, J., and Li, H. Generative adversarial network for abstractive text summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. The expressive power of neural networks: A view from the width, 2017.

Ma, W., Cheng, F., Xu, Y., Wen, Q., and Liu, Y. Probabilistic representation and inverse design of metamaterials based on a deep generative model with semi-supervised learning strategy. *Advanced Materials*, 31(35):1901111, 2019.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

Ng, A. et al. Sparse autoencoder. *CS294A Lecture notes*, 72 (2011):1–19, 2011.

Park, S., Yun, C., Lee, J., and Shin, J. Minimum width for universal approximation. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=O-XJwyoIF-k.

Perarnau, G., van de Weijer, J., Raducanu, B., and Álvarez, J. M. Invertible conditional GANs for image editing, 2016.

Pidhorskyi, S., Adjeroh, D. A., and Doretto, G. Adversarial latent autoencoders. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pp. 14104–14113, 2020.

Pinkus, A. Approximation theory of the MLP model in neural networks. *Acta numerica*, 8(1):143–195, 1999.

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. Generative adversarial text to image synthesis, 2016.

Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., and Cohen-Or, D. Encoding in style: A StyleGAN encoder for image-to-image translation. *arXiv preprint arXiv:2008.00951*, 2020.

Robertson, H. P. The uncertainty principle. *Physical Review*, 34(1):163, 1929.

Rosca, M., Lakshminarayanan, B., Warde-Farley, D., and Mohamed, S. Variational approaches for auto-encoding generative adversarial networks. *arXiv preprint arXiv:1706.04987*, 2017.

Rosca, M., Lakshminarayanan, B., and Mohamed, S. Distribution matching in variational inference. *arXiv preprint arXiv:1802.06847*, 2018.

Rudin, W. et al. *Principles of mathematical analysis*, volume 3. McGraw-hill New York, 1964.

Shen, Y. and Zhou, B. Closed-form factorization of latent semantics in GANs. *arXiv preprint arXiv:2007.06600*, 2020.

Su, J. O-GAN: Extremely concise approach for auto-encoding generative adversarial networks. *arXiv preprint arXiv:1903.01931*, 2019.

Tolstikhin, I., Bousquet, O., Gelly, S., and Schoelkopf, B. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

Villani, C. *Topics in optimal transportation*. American Mathematical Soc., 2003.

Villani, C. *Optimal transport: Old and new*, volume 338. Springer Science & Business Media, 2008.

Wold, S., Esbensen, K., and Geladi, P. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

Wolterink, J. M., Dinkla, A. M., Savenije, M. H., Seevinck, P. R., van den Berg, C. A., and Išgum, I. Deep MR to CT synthesis using unpaired data. In *International workshop on simulation and synthesis in medical imaging*, pp. 14–23. Springer, 2017.

Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., and He, X. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1316–1324, 2018.

Yi, X. and Babyn, P. Sharpness-aware low-dose CT denoising using conditional generative adversarial network. *Journal of digital imaging*, 31(5):655–669, 2018.

Yi, X., Walia, E., and Babyn, P. Unsupervised and semi-supervised learning with categorical generative adversarial networks assisted by Wasserstein distance for dermoscopy image classification. *arXiv preprint arXiv:1804.03700*, 2018.

Zha, Z.-J., Liu, J., Chen, D., and Wu, F. Adversarial attribute-text embedding for person search with natural language query. *IEEE Transactions on Multimedia*, 22 (7):1836–1846, 2020.

Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., and Metaxas, D. N. StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 5907–5915, 2017.

Zhang, H., Sindagi, V., and Patel, V. M. Image de-raining using a conditional generative adversarial network. *IEEE transactions on circuits and systems for video technology*, 2019.

Zhang, Z. and Zha, H. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM journal on scientific computing*, 26(1):313–338, 2004.

Zheng, H., Fu, J., Zeng, Y., Luo, J., and Zha, Z.-J. Learning semantic-aware normalization for generative adversarial networks. *Advances in Neural Information Processing Systems*, 33:21853–21864, 2020.

Zhu, J., Zhao, D., Zhang, B., and Zhou, B. Disentangled inference for GANs with latently invertible autoencoder. *arXiv:1906.08090v3*, 2019.