IID-GAN: an IID Sampling Perspective for Regularizing Mode Collapse

Yang Li, Liangliang Shi and **Junchi Yan**
Department of Computer Science and Engineering,
MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University
{yanglily, shiliangliang, yanjunchi}@sjtu.edu.cn

Abstract

Despite its success, generative adversarial networks (GANs) still suffer from mode collapse, i.e., the generator can only map latent variables to a partial set of modes in the target distribution. In this paper, we analyze and seek to regularize this issue with an independent and identically distributed (IID) sampling perspective and emphasize that holding the IID property referring to the target distribution for generation can naturally avoid mode collapse. This is based on the basic IID assumption for real data in machine learning. However, though the source samples $\{\mathbf{z}\}$ obey IID, the generations $\{G(\mathbf{z})\}$ may not necessarily be IID sampling from the target distribution. Based on this observation, considering a necessary condition of IID generation that the inverse samples from target data should also be IID in the source distribution, we propose a new loss to encourage the closeness between inverse samples of real data and the Gaussian source in latent space to regularize the generation to be IID from the target distribution. Experiments on both synthetic and real-world data show the effectiveness of our model.

1 Introduction

For generative models, the training data (in target space) is often assumed to be IID sampled from an unknown implicit distribution. Deep generative models often try to construct a mapping from a known distribution in source space (e.g. Gaussian distribution) to the implicit target distribution. Among popular generative schemes, e.g. Variational Auto-Encoder (VAEs) [Kingma and Welling, 2014], Generative Flow models [Rezende and Mohamed, 2015] and Generative Adversarial Networks (GANs) [Goodfellow *et al.*, 2014], GAN is widely recognized for its excellence in generating images with high resolution. Mode collapse is one of the standing issues in GAN, a phenomenon that the generator tends to get stuck in a subset of modes while excluding other parts of the target distribution [Liu *et al.*, 2020; Yang *et al.*, 2019], leading to poor generation diversity.

Efforts have been made to address mode collapse along two branches: 1) Achieve a better convergence between the generated distribution and the target (real data) distribution [Gulrajani *et al.*, 2017; Metz *et al.*, 2017]. Though the distribution convergence is recognized as the ultimate goal of generation [Goodfellow *et al.*, 2014], the implicitness of the two distributions makes the desire can only be achieved by utilizing the sampling data. Thus the goal of distribution convergence can be somehow vague and imprecise. 2) Penalize the similarity of the generated images [Elfeki *et al.*, 2019; Mao *et al.*, 2019] or apply multiple generator/discriminators [Liu and Tuzel, 2016; Nguyen *et al.*, 2017]. These methods promote the generation diversity in a more direct manner but often lack rigorous theoretical guarantees.

Existing mainstream GAN methods [Goodfellow *et al.*, 2014; Arjovsky *et al.*, 2017; Gulrajani *et al.*, 2017; Miyato *et al.*, 2018] focus on achieving the identical generated distribution as the real one. [Goodfellow *et al.*, 2014] shows that if given enough capacity and training time, the generated distribution converges to real distribution through GAN's optimization. However, due to the limited capacity of the networks (e.g. catastrophic forgetting) and the implicitness of the two distributions, the convergence only holds in theory. With the short-term guidance in training, most of the methods [Goodfellow *et al.*, 2014; Arjovsky *et al.*, 2017; Gulrajani *et al.*, 2017; Miyato *et al.*, 2018] tend to **sample-wise** boost the generation quality i.e. increase the probability density of the generated sample obeying the target distribution. Taking GAN as an example, discriminator D distinguishes whether every generation is sampled from the identical target distribution, and generator G tends to generate samples with high discriminative confidence of the current D. While in VAE-based or Flow-based approaches [Kingma and Welling, 2014; Rezende and Mohamed, 2015], log-likelihood is applied to increase the density of every generation. These methods rarely study the coupled relation between the generated samples to constitute a distribution, and each step the optimized sample is pulled to the center of the distribution to maintain high probability density, while the characteristics of the overall distribution e.g. variance are not explicitly controlled. This leads them to merely improve the quality of the individual generated samples, trivializing to maintain diversity as real data does. This may stem from the dilemma that a single sample can hardly estimate a distribution through nets, unless the nets possess unrestricted capability and do not exhibit catastrophic forgetting [McCloskey and Cohen, 1989].

In this paper, rather than study the relation between distributions, we directly optimize the sampling relation between **generated sample batches** and the target distribution. We first make a basic yet under-exploited (in GAN literature) observation that datasets for generation are assumed to be independent and identically distributed (IID) sampled from an unknown target distribution (i.e. real distribution). We aim to achieve that the generated samples are **independently** sampled from the **identical** target distribution. If the latents $\{\mathbf{z}\}$ are IID sampled from the source, it is guaranteed that the generations $\{G(\mathbf{z})\}$ are IID samples from the generated distribution, but not necessarily IID from the target. Note the independence here is considered from the perspective of statistical test, which needs consistency over samples $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ and the target distribution (or between the latents $\{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n\}$ and source distribution). Sample batches can contain relatively more complete features characterizing a distribution, improving optimization efficiency.

To address mode collapse and get IID generation, we first define *mode completeness* as the key requirement of the *ideal* mapping from the source to the target. As the target distribution is implicit and we can only obtain its finite IID samples, we propose a weaker/necessary condition[1] for IID generation based on an inverse mapping: if the real data are IID samples from target distribution, then their inverses in the source space are also IID. To achieve IID property, a common and straightforward idea is to drive the distribution of the overall inverse samples close to a standard Gaussian by certain measures. This idea differs from existing inversion-based methods [Srivastava *et al.*, 2017; Donahue *et al.*, 2017; Yu *et al.*, 2019] which learn the relationship between the individual latent sample $\mathbf{z}$ and real data $\mathbf{x}$ through the discriminator, yet the relationship among $\{\mathbf{z}\}$ is not fully exploited. The highlights of the paper are:

1) We take an IID perspective to mode collapse in GAN. The requirement for an ideal generator is proposed that it should satisfy the so-called mode completeness by Def. 1. Limited by the finiteness of real samples, we resort to a necessary condition of mode completeness in Prop. 1, which requires the IID property of the inverse samples from the real data.

2) Guided by Prop. 1, a regularizer is devised to avoid mode collapse. We enforce the inverse sample batch to be distributed close to a standard Gaussian by Wasserstein distance, termed as Gaussian consistency loss (see Sec. 4.3). QQ-plot, Shapiro–Wilk and Kolmogorov-Smirnov statistics are used to test the IID property of the inverse Gaussian samples.

3) We show that IID-GAN outperforms baselines by different metrics on synthetic data w.r.t. the number of covered modes, quality and reverse KL divergence. Our regularization technique also performs competitively on natural images. Unsupervised disentanglement feature learning and conditional generation are also investigated.

2 Related Work

Since its debut [Goodfellow *et al.*, 2014], subsequent works of GANs have emerged to improve training stability and generation quality. However, issues still remain for GANs, and mode collapse is one of the most common challenges. In this section, we introduce methods that are directly or indirectly dedicated to address mode collapse.

Improving training behavior. The promotion of distribution convergence can naturally avoid mode collapse. Unrolled GAN [Metz *et al.*, 2017] presents a surrogate objective to train the generator, along with an unrolled optimization of the discriminator to improve training stability. Wasserstein GANs [Arjovsky *et al.*, 2017] and its variants e.g. [Gulrajani *et al.*, 2017] modify the optimization objective to stabilize training. Although these methods all claim to improve the convergence between the generated and real distribution, the convergence focus more on generation quality that they tend to align every generated sample closer to the real distribution but focus less on the inter-sample relation.

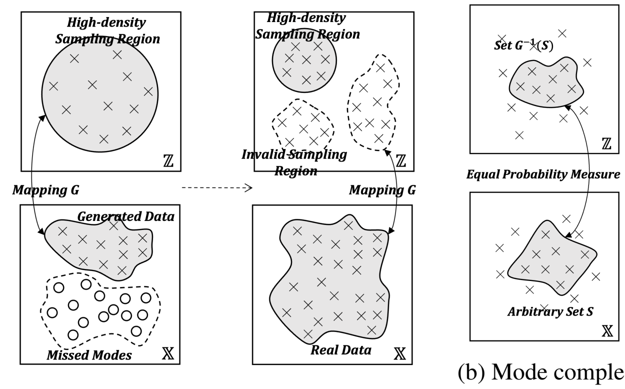
Enforcing to capture diverse modes. Many other methods address mode collapse by penalizing similarity. GDPP [Elfeki *et al.*, 2019] applies the determinantal point process theory, which imposes a penalty on the discriminator to enforce the convergence of the covariance matrix between the features of generated samples and real data. BuresGAN [De Meulemeester *et al.*, 2021] matches the discriminator's last layer output through Bures distance. AdvLatGAN [Li *et al.*, 2022] mines sample pairs that tend to collapse and regularize them to maintain diversity. All these methods are concerned with the relationship among generated results. However, these methods lack stability guarantees for diverse generation, as they ignore that the generated data shall statistically maintain the same diversity as the real distribution.

Multiple generators and discriminators. Involving multiple generators to achieve wider coverage over the real distribution can also benefit diverse generation. In [Liu and Tuzel, 2016], two coupled generators are trained with parameter sharing to learn the real distribution. The multi-agent system MAD-GAN [Ghosh *et al.*, 2018] involves multiple generators along with one discriminator, and it implicitly encourages each generator to learn a partial set of modes. On the other hand, multiple discriminators are used in [Durugkar *et al.*, 2017] as an ensemble. AMAT [Mangalam and Garg, 2021] introduce a novel training procedure that adaptively spawns additional discriminators to remember previous modes of generation. These methods directly seek to utilize more networks to capture more modes, but omit to reach the essence of mode collapse.

Learn the representations by inverse mapping. BiGAN [Donahue *et al.*, 2017], VEEGAN [Srivastava *et al.*, 2017], VAE-based models [Kingma and Welling, 2014] and recently proposed MGGAN [Bang and Shim, 2021], DistGAN [Tran *et al.*, 2018] involve encoding networks (inverse mapping) of the generator to encourage the convergence between the inverse distribution of real data and source distribution. Flow-GAN [Grover *et al.*, 2018] utilizes a flow-based generator that allows for invertibility. However, it requires very special network design that can significantly limit the network's expressive power and inference speed.

However, similar to the previous discussion about methods improving training behavior, the inverse mapping procedure ignores the IID requirement, i.e., they merely sample-wise boost the generation quality (e.g. the probability of the sample locating in real data distribution) and fail to explicitly control

[1] Note that it will become a necessary and sufficient condition if there are an infinite amount of real data.



(a) Mode collapse caused by real data's inverses outside high-density sampling region.

(b) Mode completeness w/ equal probability measures

Figure 1: Mapping from source to target: (a) Left: high-density samples can only be mapped to part of real data i.e. mode collapse; Right: only part of real samples' inverses lie in high-density sampling region, and the inverses outside the region can hardly be sampled, leading to mode collapse. (b) Assuming the existence of the inverse of mapping $G(\cdot)$, to solve mode collapse and get IID generation, the probability measures of the set $G^{-1}(S)$ and S should be equal for any set S in target space. See Fig. 3 for experimental examples.

the coupled relation of samples to constitute a distribution. For example, VEEGAN seeks to discriminate real/fake $(\mathbf{z}, \mathbf{x})$ pairs by the discriminator while the loss is enforced to each pair. As the networks exhibit limited capacity e.g. catastrophic forgetting, the discriminator cannot accurately characterize the implicit distribution through a sequence of single sample learning, which hinders the convergence between the inverse and source distributions as shown in Fig. 3. Under the guidance of IID generation, we take one step further to reveal the impact of IID informed inverse mapping on solving mode collapse. Specifically, Prop. 1 calls for the entire inverse sample batch $\{G^{-1}(\mathbf{x})\}$ to be IID sampled from Gaussian, which requires considering the entire sample set, regularizing the optimization with more samples.

3 Preliminaries and Motivation

Fig. 1a presents a basic interpretation of mode collapse on top of the mapping between the source and target spaces. To address mode collapse, Fig. 1b shows a probability measure requirement for source and target space samples (i.e. probability measures for set-set pairs in the two spaces should be equal). We first define *mode completeness* corresponding to the ideal generative mapping between two probability measures:

Definition 1. **(Mode Completeness)** *The probability measures α and β are defined in the source space $\mathcal{A}$ and the target space $\mathcal{B}$, respectively. The generative mapping $G : \mathcal{A} \to \mathcal{B}$ is defined as mode completeness from α to β if G satisfies:*

$$\beta(\mathcal{S}) = \alpha(\mathbf{z} \in \mathcal{A} : G(\mathbf{z}) \in \mathcal{S}\}), \tag{1}$$

where $\mathcal{S} \subset \mathcal{B}$ is an arbitrary set in the the target space.

Eq. 1 defines the same operation of the push-forward operator [Peyré *et al.*, 2019] $\beta = G_{\#}\alpha$ in optimal transportation. So the mode completeness can also be interpreted as regularizing the mapping G such that $\beta = G_{\#}\alpha$, and the operator $G_{\#}$ implies that G pushes forward the mass of α to β [Peyré *et al.*, 2019]. Based on Def. 1, mode collapse can be naturally avoided because the values of the probability measures given the corresponding sets based on G are equal (i.e. $\alpha(\{\mathbf{z}_i\}) = \beta(\{T(\mathbf{z}_i)\})$). Then given the equal probability measures, when $\mathbf{z}_i$ are IID sampled from α, the generations $\{G(\mathbf{z}_i)\}$ are IID samples from β.

Mode completeness $\beta = G_{\#}\alpha$ can hardly be completely achieved since β is complex and implicit. Considering the basic assumption that the real training data are IID sampled from β, if we assume the existence of $G^{-1} : \mathcal{B} \to \mathcal{A}$ such that $z = G^{-1}(G(z))$ and $x = G(G^{-1}(x))$ for any $z \in \mathcal{A}$ and $\mathbf{x} \in \mathcal{B}$, then we can obtain a necessary condition for mode completeness. Such a necessary condition for IID generation is formalized Proposition 1:

Proposition 1. **(IID for Inverse of Target Samples)** *If the mapping G satisfies mode completeness from the source probability measure α to the target measure β and its inverse G^{-1} exists, then given IID samples $\{\mathbf{x}^{(i)}\}_{i=1}^{n}$ from β, their inverses $\{G^{-1}(\mathbf{x}^{(i)})\}_{i=1}^{n}$ can be viewed as IID samples from α.*

To prove Proposition 1, we first provide a proof for Lemma 1, which represents a more general scenario.

Lemma 1. *Assume the existence of mapping T which satisfies $T_{\#}\alpha = \beta$ and the existence of its inverse T^{-1}. Let ρ be the joint probability measure of n independent copies of β, denoted as $\beta_1, \cdots, \beta_n$ and π is the joint probability measure of n independent copies of α, denoted as $\alpha_1, \cdots, \alpha_n$. Then we have $\tilde{T}_{\#}\pi = \rho$, where $\tilde{T}$ is the concatenation of n mapping Ts that satisfies $\tilde{T}(x_1, \cdots, x_n) = [T(x_1), \cdots, T(x_n)]$. Then we can obtain that π is the joint probability measure with n* ***independent*** *probability measure $\alpha_1, \cdots, \alpha_n$.*

Proof. Given n measurable set $\mathcal{S}_i \subset \mathcal{B}$ and $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \times \mathcal{S}_n$, with the independence between $\{\beta_i\}_{i=1}^{n}$, we obtain:

$$\rho(\mathcal{S}) = \beta_1(\mathcal{S}_1)\beta_2(\mathcal{S}_2)\cdots\beta_n(\mathcal{S}_n) \tag{2}$$

Since $\tilde{T}_{\#}\pi = \rho$ and $T_{\#}\alpha = \beta$, we know that $\rho(\mathcal{S}) = \pi(\tilde{T}^{-1}(\mathcal{S}))$ and $\beta(\mathcal{S}_i) = \alpha(T^{-1}(\mathcal{S}_i))$. Then we have:

$$\pi(\tilde{T}^{-1}(\mathcal{S})) = \alpha_1(T^{-1}(\mathcal{S}_1))\cdots\alpha_n(T^{-1}(\mathcal{S}_n)) \tag{3}$$

which indicates that π is the joint probability measure with **independent** $\alpha_1, \cdots, \alpha_n$. □

The proof of Proposition 1 can be easily obtained from Lemma 1 by setting n measurable sets as $\mathcal{S}_1 = \{x_1\}, \cdots, \mathcal{S}_n = \{x_n\}$ where $x_1, \cdots, x_n$ are n IID samples from target distribution. According to Eq. 3, we obtain

$$\alpha(T^{-1}(x_1))\cdots\alpha(T^{-1}(x_n)) = \pi(\tilde{T}^{-1}(\{x_1, \cdots, x_n\})) \tag{4}$$

which implies that $\{T^{-1}(x^{(i)})\}_{i=1}^{n}$ can be viewed as n independent samples from source distribution.

Since it is commonly assumed that the real data $\{\mathbf{x}^{(i)}\}_{i=1}^{n}$ are independently sampled from an unknown distribution β, their inverses $\{G^{-1}(\mathbf{x}^{(i)})\}_{i=1}^{n}$ can be viewed as IID samples from a known distribution α. To satisfy mode completeness, we regularize the inverse samples of real data to be closer to the IID samples from α. Since we can hardly obtain a strict inverse mapping G^{-1}, we calls for an approximation of G^{-1} as F, which is obtained by penalizing:

$$\mathbf{z} = F(G(\mathbf{z})) \qquad \mathbf{x} = G(F(\mathbf{x})). \tag{5}$$

4 The Proposed Approach

Our loss mainly consists of three parts as shown in Fig. 2.

4.1 Adversarial Learning Term: GAN Loss

The vanilla GAN [Goodfellow *et al.*, 2014] consists of a discriminator $D : \mathcal{R}^d \to \mathcal{R}$ and a generator $G : \mathcal{R}^M \to \mathcal{R}^d$, which are typically embodied by deep neural networks. Given the empirical distribution $p(x)$, D distinguishes whether the generated sample is from real data, while G models the mapping from Gaussian sample $\mathbf{z}$ to a target space sample. The objective $V(G, D)$ is optimized for the discriminator and generator by solving the min-max:

$$E_{\mathbf{x}\sim p_r(\mathbf{x})}\left[\log(D(\mathbf{x}))\right] + E_{\mathbf{z}\sim p(\mathbf{z})}\left[\log(1 - D(G(\mathbf{z})))\right] \tag{6}$$

The first term denotes the probability expectation of $\mathbf{x}$ coming from the real data distribution $p(x)$ and the second term involves the input distribution $p(\mathbf{z})$, which is embodied in this paper as a standard multi-dimensional (M-D) Gaussian distribution $\mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$. $\mathbf{I} \in \mathcal{R}^{M\times M}$ denotes the identity matrix.

4.2 Reconstruction Term: Cycle-Consistency Loss

Prop. 1 calls for the existence of the generator's inverse, i.e. G^{-1}. To achieve the approximation for the holding of $\mathbf{z} = F(G(\mathbf{z}))$ and $\mathbf{x} = G(F(\mathbf{x}))$ where F is also a network-based mapping, a cycle-consistency loss can be an effective solution [Srivastava *et al.*, 2017; Donahue *et al.*, 2017]:

$$L_{re}(G, F) = \underbrace{E_{\mathbf{z}}\|\mathbf{z} - F(G(\mathbf{z}))\|_1}_{\text{to achieve inverse mapping}} + \lambda \underbrace{E_x\|\mathbf{x} - G(F(\mathbf{x}))\|_1}_{\text{to guarantee the existence}} \tag{7}$$

Here $\lambda = d/M$ is the dimension ratio of $\mathbf{x}$ to $\mathbf{z}$. The first term promotes F to be the inverse of G, which takes the reconstruction loss as the expectation of the cost for auto-encoding noise vectors [Srivastava *et al.*, 2017]. The second term promotes $F(\mathbf{x}) \in \mathcal{R}^M$, which makes $\tilde{\mathbf{z}} = F(\mathbf{x})$ possible to be sampled from Gaussian. Then for each $\mathbf{x}$, we can obtain the corresponding $\tilde{\mathbf{z}}$ in $\mathcal{R}^M$ satisfying $G(\tilde{\mathbf{z}}) = \mathbf{x}$, which guarantees the existence of the real samples' inverses in source space. Note that it does not offer any help to avoid mode imbalance, since the imbalance refers to the overall distribution rather than the individual data points studied here.

4.3 Regularizer: Inverse Distribution Consistency

Recall the necessary condition for IID generation proposed in Prop. 1: assume that the given real data $\{\mathbf{x}^{(i)}\}_{i=1}^n$ are IID sampled from $p(x)$, then the inverse samples of the real data will be independent and obey the same distribution $p(\mathbf{z})$ (i.e. IID samples from source distribution). Thus, given a batch of real data, $\{F(\mathbf{x}^{(i)})\}_{i=1}^n$ should be closer to independent samples from standard Gaussian $p(\mathbf{z})$.

M-D Gaussian consistency loss. Suppose the inverse samples $\{\tilde{\mathbf{z}}^{(i)}\}_{i=1}^n \in \mathcal{R}^M$ of real data obey the Gaussian distribution $\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ in latent space, then the maximum likelihood estimation can be formulated as:

$$\tilde{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n}\tilde{\mathbf{z}}^{(i)}, \quad \tilde{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n}\left(\tilde{\mathbf{z}}^{(i)} - \tilde{\boldsymbol{\mu}}\right)^{\top}\left(\tilde{\mathbf{z}}^{(i)} - \tilde{\boldsymbol{\mu}}\right) \tag{8}$$

where $\tilde{\boldsymbol{\mu}} \in \mathcal{R}^M$, and $\tilde{\boldsymbol{\Sigma}} \in \mathcal{R}^{M\times M}$ is the estimated covariance. Our goal is to make the Gaussian $q(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}})$ closer to the standard Gaussian $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$, based on which we can view $\{\tilde{\mathbf{z}}^{(i)}\}_{i=1}^n$ as IID samples from standard Gaussian $p(\mathbf{z})$. To this end, we introduce a distribution closeness loss L_{Gau}. For example, it can be specified as the square of Wasserstein distance of two Gaussians:

$$L_{Gau} = \|\tilde{\boldsymbol{\mu}}\|^2 + trace(\tilde{\boldsymbol{\Sigma}} + \mathbf{I} - 2\tilde{\boldsymbol{\Sigma}}^{1/2}) \tag{9}$$

The two Gaussian distributions $q(\mathbf{z})$ and $p(\mathbf{z})$ can also be evaluated by static divergence, e.g., p-norm, KL-divergence. More details about loss selections are presented in Appendix A.

Decoupling M-D Gaussian into M 1-D Gaussians. For large M and small batch size, the training may potentially suffer from the curse of dimensionality for estimating $\boldsymbol{\Sigma}$. So we also devise a simplified consistency loss by decoupling the M-D Gaussian loss to the sum of the 1-D Gaussian losses of M ones. This design introduces an assumption on the covariance that the non-diagonal values are all zero. Equivalently, the inverse samples $\{\tilde{\mathbf{z}}^{(i)}\}_{i=1}^n$ follow M 1-D Gaussian $\mathcal{N}(\tilde{\mathbf{z}}; \boldsymbol{\mu}, \boldsymbol{\sigma})$ and then we can get the estimation $\tilde{\boldsymbol{\mu}}$ and $\tilde{\boldsymbol{\sigma}}$.

Our goal is to make the 1-D Gaussian distribution $q(\mathbf{z}_j)$ closer to the standard 1-D Gaussian distribution in each dimension j. Similar to the M-D case, we can design the Gaussian loss with Wasserstein distance between two Gaussian distributions, and summing over M dimensions:

$$L_{Gau} = \sum_{m=1}^{M}\left(\tilde{\boldsymbol{\mu}}_m^2 + (\tilde{\boldsymbol{\sigma}}_m - 1)^2\right) \tag{10}$$

Throughout the rest of this paper, we directly call it the Gaussian loss, omitting the term *consistency*.

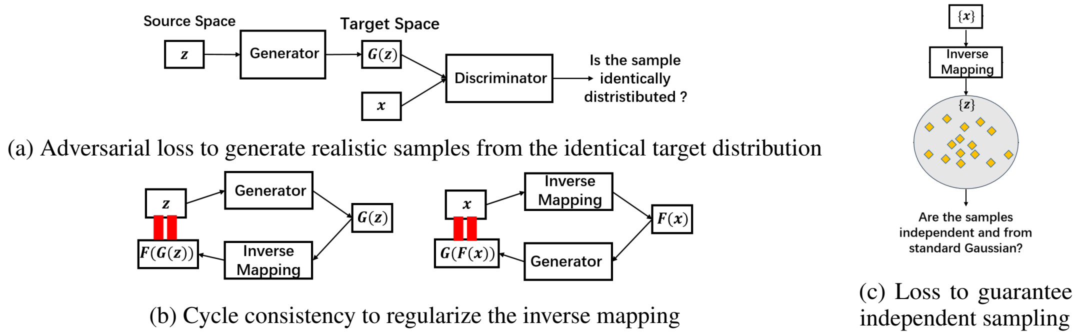
4.4 IID-GAN: an Overview

By inputting samples from $p(\mathbf{z})$ together with the corresponding inverse samples $\{\tilde{\mathbf{z}}\}$ from the real data, we optimize G, D, and F by adversarial learning to push $\{\tilde{\mathbf{z}}\}$ closer to the sampling results of $p(\mathbf{z})$. Then we can obtain the final loss as:

$$\min_{G,F}\max_{D} V(G, D) + \lambda_{re}L_{re}(G, F) + \lambda_{Gau}L_{Gau}(F) \tag{11}$$

where $L_{Gau}(F)$ can be specified according to different distances or divergences and $\lambda_{re}, \lambda_{Gau}$ are weight parameters, which will be discussed in detail in the experiment section.

Remarks for Conditional Generation. IID-GAN can be extended to the conditioned case [Mirza and Osindero, 2014] when the label of the real data c is known. As for conditional IID-GAN, $(\mathbf{z}, c)$ are the inputs of the generator G and the real $\mathbf{x}$ is the input of F for classification. The Gaussian loss is used



(a) Adversarial loss to generate realistic samples from the identical target distribution

(b) Cycle consistency to regularize the inverse mapping

(c) Loss to guarantee independent sampling

Figure 2: Generator G maps M-D Gaussian samples to targets and F inverts the target back to a source sample obeying M-D Gaussian.

to maintain the independence condition and learn the diversity of hidden features (e.g. the thickness for MNIST).

Remarks for the Disentanglement View. Many previous studies [Higgins *et al.*, 2017; Kim and Mnih, 2018] learn the unsupervised disentanglement representation with the assumption of independent factors, i.e. $q(\mathbf{z}) = \prod_{j=1}^{M} q(\mathbf{z}_j)$. Our work presents a new viewpoint with an M-D Gaussian guarantee. When $q(\mathbf{z})$ approximates an M-D standard Gaussian, it is obvious that $\mathbf{z}$ is independent for different dimensions. However, varying $\mathbf{z}_i$ can not disentangle different modes as shown in the second column of Fig. 3. We observe that representing the data in polar coordinates and varying the polar angle and diameter can achieve better performance for unsupervised disentangle representations.

IID testing for Gaussian. The regularization assumes a non-standard Gaussian and enforces the consistency of two Gaussians, which raises the doubt: Is the consistency useful for IID? We apply mathematical statistics here to test the Gaussian IID property. Specifically, we adopt the QQ-plot, Shapiro–Wilk test (SW) [Shapiro and Wilk, 1965], Kolmogorov-Smirnov test (KS) [Gosset, 1987] to show whether the samples are IID sampled from standard Gaussian. The results are shown in Fig. 5 and Table 2.

5 Experiments and Discussion

Experiments are conducted on a single GeForce RTX 3090. Synthetic data results are performed on GeForce RTX 2080Ti.

5.1 Experiments on Synthetic Datasets

Since the distribution is known for synthetic data, mode collapse can be directly measured. In line with [Metz *et al.*, 2017], we simulate two synthetic datasets: **i) Ring dataset.** It consists of a mixture of 8 2-D Gaussians $p(\mathbf{z})$ with mean $\{(2\cos(i\pi/4), 2\cos(i\pi/4))\}_{i=1}^{8}$ and standard deviation 0.001. 12.5K samples are simulated from each Gaussian (100K samples in total). **ii) Grid dataset.** It consists of a mixture of 25 2-D isotropic Gaussians i.e. $p(\mathbf{z})$ with mean $\{(2i, 2j)\}_{i,j=-2}^{2}$ and standard deviation 0.0025. 4K samples are simulated from each Gaussian (100K samples in total).

Metrics and network architecture. Following [Metz *et al.*, 2017; Elfeki *et al.*, 2019], we adopt the number of covered modes, generation quality[2] and reverse KL divergence as the evaluation metrics. Since in the experiment, each mode shares the same number of real samples, one can calculate the reverse KL divergence between the generated distribution and the real one [Nguyen *et al.*, 2017]. Note the reverse KL divergence is not strictly defined, as $\sum_{i=1}^{m} p_i < 1$ (there exist invalid generated points), thus it allows being negative. We adopt network architectures consisting of three linear layers with hidden dimensions of 100, 200, and 100, along with ReLU activation, for both the generator and discriminator.

Models	2D-Ring			2D-Grid		
	#Mode↑	Quality% ↑	RKL↓	#Mode↑	Quality% ↑	RKL↓
GAN	3.6 ± 0.5	98.8 ± 0.6	0.92 ± 0.11	18.4 ± 1.6	$\mathbf{98.0} \pm \mathbf{0.4}$	0.75 ± 0.25
BiGAN	6.8 ± 1.0	38.6 ± 9.5	0.43 ± 0.18	24.2 ± 1.2	83.4 ± 2.9	$\mathbf{0.26} \pm 0.20$
Unrolled GAN	6.4 ± 2.2	98.6 ± 0.5	0.42 ± 0.53	8.2 ± 1.7	98.7 ± 0.6	1.27 ± 0.17
VEEGAN	5.4 ± 1.2	38.8 ± 16.7	0.40 ± 0.10	20.0 ± 2.6	85.0 ± 5.9	0.41 ± 0.10
IID-GAN (1-D)	8.0 ± 0.0	97.3 ± 0.6	0.18 ± 0.06	25.0 ± 0.0	97.8 ± 0.49	0.32 ± 0.09
IID-GAN (M-D)	$\mathbf{8.0} \pm \mathbf{0.0}$	$\mathbf{99.0} \pm \mathbf{0.2}$	$\mathbf{0.17} \pm \mathbf{0.06}$	$\mathbf{25.0} \pm \mathbf{0.0}$	$\mathbf{98.0} \pm \mathbf{0.4}$	$\mathbf{0.26} \pm \mathbf{0.12}$

Table 1: Results for Ring and Grid synthetic datasets.

Quantitative results. The comparison involve vanilla GAN [Goodfellow *et al.*, 2014], BiGAN [Donahue *et al.*, 2017], Unrolled GAN [Metz *et al.*, 2017] and VEEGAN [Srivastava *et al.*, 2017] on Ring and Grid datasets in Table 1. IID-GAN variants are the only methods covering all modes on both datasets and IID-GAN (M-D) achieves the best quality and RKL performance compared to other peer methods.

Visualization with Gaussian inverse samples. Models involving inverse mappings (i.e. encoders) allow the visualization for the inverses of real samples. Fig. 3 present the visualization results including the inverse from real data, source Gaussian samples and generated results of VAE [Kingma and Welling, 2014], VEEGAN, BiGAN and IID-GAN variants. As shown in the left four columns, VAE can cause the overlap of the inverse samples $\mathbf{z}$, which may lead to bad generations (white points). BiGAN and VEEGAN learn the relations between $\mathbf{z}$ and $\mathbf{x}$ rather than the relation among the samples $\{G^{-1}(\mathbf{x})\}$, which leads to the failure of constructing 2D inverse Gaussian samples as shown in the first column. For IID-GAN, as shown in the fifth column for IID-GAN (M-D), the inverse samples are very similar to the Gaussian samples, which show the effectiveness of the regularization.

Visualization of generated distribution. Fig. 4 shows the generation results of IID-GAN and peer methods. IID-GAN

[2] We follow [De Meulemeester *et al.*, 2021]: if the generated data is within 3 times std of the Gaussian, consider it a valid generation (otherwise bad). The resulting ratio is used as the generation quality.

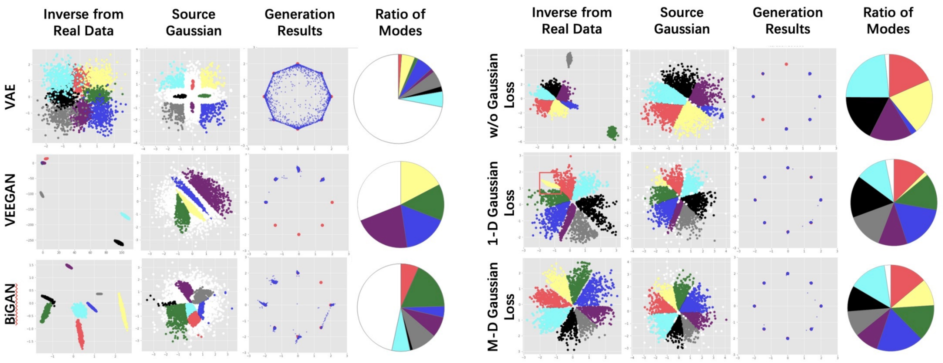
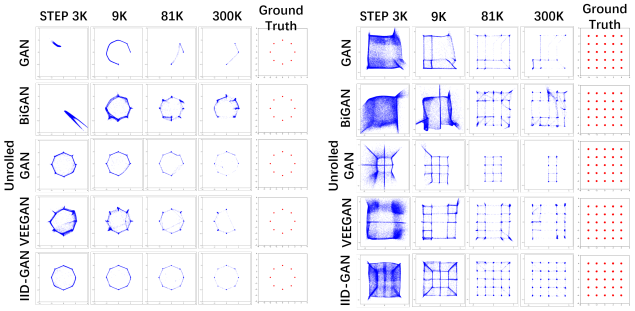


Figure 3: Example for 2-D source ($\mathbf{z}$) to 2-D target ($\mathbf{x}$) generation and inverse adopting 8-mode Ring dataset as the real training set. Left 4 columns are the results of VAE, BiGAN, VEEGAN, while the right 4 columns are the results of IID-GAN under different Gaussian consistency losses as detailed in Sec. 4.3 after training 24K batches. For each half part, column 1 and column 5 show the inverse of the real target data, column 2 and column 6 show the sampled $\mathbf{z}$ from Gaussian in source space. Source points are in nine colors (8 'modes' + 1 'bad') according to their generation's mode in the target space. The pie charts show the ratio of valid generation points in different modes.



(a) Results of Ring data

(b) Results of Grid data

Figure 4: Generation results on Ring and Grid datasets. Blue: generated samples; Red: ground truth distribution.

significantly outperforms other peer methods in terms of mode coverage and better fits the target distribution.

IID test for inverse samples. We support IID test for Gaussian distribution on Ring dataset in Table 2. Shapiro-Wilk (SW) and Kolmogorov-Smirnov (KS) Statics are calculated to verify IID property. In Fig. 5, QQ-plot are supported for IID test. Compared to other peer methods, IID-GAN's plots are closer to the red diagonal, implying that the inverse samples of IID-GAN are closer to Gaussian.

Ablation study for Gaussian consistency loss. The right four columns of Fig. 3 compare the results of 1-D, M-D and without Gaussian consistency loss in Ring datasets. For 1-D loss's inverse samples in source space, the yellow sample region in the red box leads to a tendency to mode collapse, while the generated modes in the M-D case are more uniformly scaled and more effective for solving mode collapse.

5.2 Experiments on Real-world Data

The experimented image datasets include MNIST [LeCun *et al.*, 1998], Stacked MNIST [Metz *et al.*, 2017], CIFAR-10 [Krizhevsky *et al.*, 2009], STL-10 [Coates *et al.*, 2011], LSUN [Yu *et al.*, 2015] and CELEBA [Liu *et al.*, 2015]. We adopt different architectures to evaluate our model, mainly following the previous studies [Radford *et al.*, 2016; De Meulemeester *et al.*, 2021; Miyato *et al.*, 2018]. All the compared models are trained in 100K steps. Details about the network architectures are presented in Appendix B.2. We generally adopt p-norm distance as M-D Gaussian loss. Please refer to Appendix B.3 for more details.

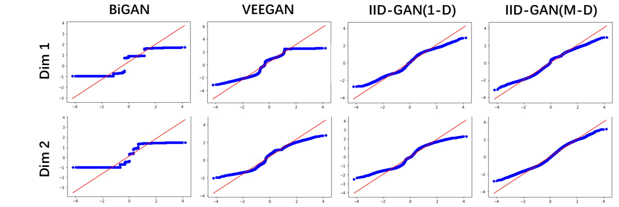
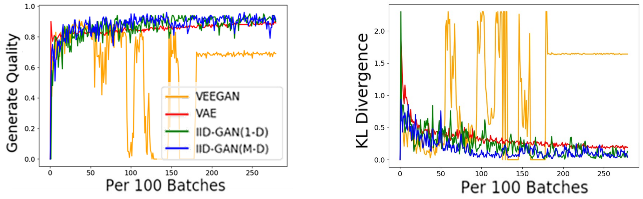


Figure 5: QQ-plot of standard 2-D Gaussian over two dimensions. The closer to the diagonal, the closer to the Gaussian distribution.



(a) Generation Quality

(b) KL divergence

Figure 6: Generation quality and KL divergence (for diversity) calculated through the generations' classification results.

Metrics. We use Inception Score (IS) [Salimans *et al.*, 2017] and Fréchet Inception Distance (FID) [Heusel *et al.*, 2017]. Following [Richardson and Weiss, 2018], JSD measures the distribution distance between generated images and real images through clustering, which generally capture the distribution of image modes, thus it is more diversity related. For easily distinguishable datasets like MNIST, we use the number of covered modes and reverse KL for evaluation.

Training stability. In Fig. 6, the generation quality and KL divergence are evaluated on MNIST. The generation quality

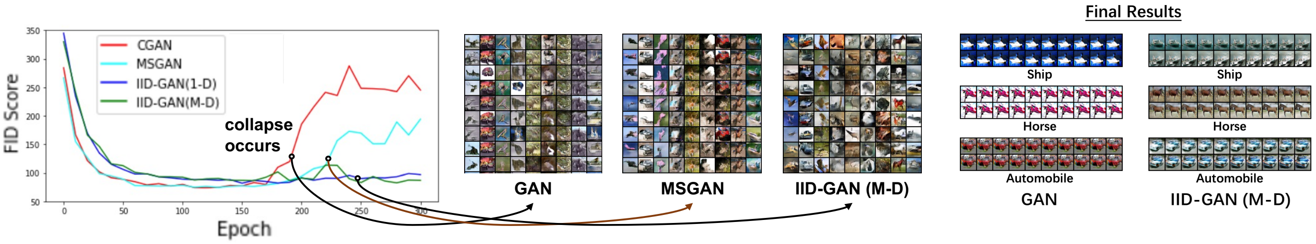


Figure 7: Conditional results on CIFAR-10 for CGAN, MSGAN and conditional IID-GAN. The generation results of CGAN and MSGAN can easily deteriorate as training proceeds in terms of FID score, while IID-GAN performs consistently well. The final results are achieved by changing one dimension of the input latent code z by increasing or decreasing the value equidistantly, producing a gradual series of images.

Model	SW Static ↑	KS Static ↓	
		$dim1$	$dim2$
BiGAN	0.8537	0.3900	0.1637
VEEGAN	0.9567	0.3141	0.2350
IID-GAN (1-D)	0.9548	**0.0882**	0.0866
IID-GAN (M-D)	**0.9824**	0.1185	**0.0579**

Table 2: SW and KS Static evaluation for IID test on Ring.

Model	**Stacked MNIST**		
	#Mode↑	KL↓	FID↓
GAN	392.0	8.012	97.788
VEEGAN	761.8	2.173	86.689
PACGAN	992.0	0.277	117.128
IID-GAN (1-D)	996.4	0.152	86.911
IID-GAN (M-D)	**999.7**	**0.101**	**69.675**

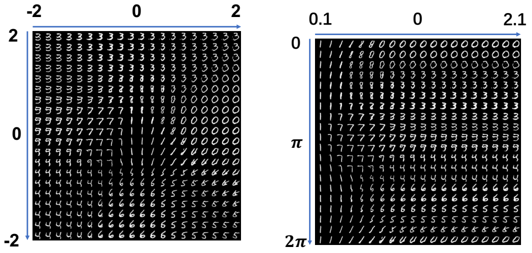
Table 3: Results on Stacked MNIST.

Model	**CIFAR-10**			**STL-10**		
	JSD×10^2↓	IS↑	FID↓	JSD×10^2↓	IS↑	FID↓
Unrolled GAN	3.23±0.36	7.28±0.13	27.07±0.74	3.32±0.27	8.28±0.03	38.79±0.56
VEEGAN	4.83±0.33	7.12±0.07	26.94±0.94	5.23±0.36	8.39±0.05	37.25±0.32
MAD-GAN	7.53±0.78	7.00±0.02	29.24±0.36	9.15±0.43	7.99±0.03	40.03±0.10
GDPP	5.25±0.29	7.23±0.06	26.90±0.15	4.18±0.19	8.36±0.02	38.12±0.22
BuresGAN	6.19±0.72	7.22±0.01	28.28±0.28	4.33±0.19	8.17±0.03	39.22±0.39
IID-GAN	**2.95±0.22**	**7.35±0.03**	**25.71±0.28**	**3.11±0.25**	**8.43±0.04**	**36.16±0.51**

Table 4: Results on CIFAR-10 and STL-10.

is defined as the ratio of high-confidence samples evaluated by a trained classifier. 1-D and M-D IID-GAN can lead to less mode imbalance (which is evaluated by KL divergence) and higher quality than VAE, while VEEGAN is unstable and often fails to successfully generate in case of bad initialization.

Quantitative results. **(i)** Table 3 show the #Mode, KL divergence and FID results on Stacked MNIST. The KL divergence calculation is based on the classification results by the classifier proposed by [Dieng *et al.*, 2019]. The architecture is DCGAN-based in line with [Radford *et al.*, 2016]. **(ii)** Table 4 shows the experimental results of CIFAR-10 [Krizhevsky *et al.*, 2009] and STL-10 [Coates *et al.*, 2011]. We adopt JSD, IS, FID as the metrics, including UnrolledGAN [Metz *et al.*, 2017], VEEGAN [Srivastava *et al.*, 2017], MAD-GAN [Ghosh *et al.*, 2018], GDPP [Elfeki *et al.*, 2019] and BuresGAN [De Meulemeester *et al.*, 2021] for comparison. To fit with advanced GAN achievements, we adopt SNGAN [Miyato *et al.*, 2018] as the backbone. **(iii)** We also evaluate on larger scale datasets e.g. CELEBA [Liu *et al.*, 2015] and LSUN Church [Yu *et al.*, 2015] with more recently proposed works pursuing diversity for comparison including Dist-GAN [Tran *et al.*, 2018], MGGAN [Bang and Shim, 2021] and AMAT [Mangalam and Garg, 2021]. We adopt FID, JSD as the evaluation metrics. The results are presented in Table 5. **(iv)** We evaluate IID-GAN with StyleGAN2-ada [Karras *et al.*, 2020] backbone on 64×64 resized AFHQ, LSUN via FID and KID metrics for same training iterations, i.e. 10M images trained (measured in real images shown to the discriminator following StyleGAN2-ada) and show its positive effect.



(a) Cartedian Coordinate

(b) Polar Coordinate

Figure 8: Disentanglement by uniformly sampling on MNIST.

Method	**CELEBA**		**LSUN**	
	FID↓	JSD×10^2↓	FID↓	JSD×10^2↓
Dist-GAN	37.374±0.994	3.604±0.202	34.491±0.487	5.042±0.372
MGGAN	19.491±0.348	3.507±0.159	13.018±0.364	7.464±0.805
AMAT	29.720±0.728	4.789±0.158	31.585±0.923	4.914±0.084
IID-GAN	**18.715±0.130**	**3.294±0.183**	**11.755±0.451**	**4.210±0.651**

Table 5: Evaluation (calculated at 100K steps of training) on real-world CELEBA and LSUN, using SNGAN as backbone.

Method	**AFHQ Cat**		**LSUN Church**	
	FID↓	KID×10^3↓	FID↓	KID×10^3↓
StyleGAN2-ada	6.386±0.503	1.077±0.113	3.530±0.021	1.415±0.008
IID-GAN	**5.373±0.142**	**0.897±0.126**	**3.208±0.111**	**1.317±0.095**

Table 6: Evaluation on StyleGAN2-ada backbone.

Results on conditional generation and disentanglement. Fig. 7 shows the results for conditional generation on CIFAR-10. The conditional IID-GANs are the most stable and robust model, and maintain the most diverse generation for each given category. Fig. 8 shows the visualization results in Cartesian and Polar Coordinate on MNIST. To better visualize the distribution, we use a 2-D latent space for generation.

6 Conclusion

To address GAN's long-standing mode collapse issue, we provide an IID sampling perspective to analyze the generation behavior and offer our new methodology guidance, which is orthogonal to existing literature. The proposed IID-GAN shows its effectiveness on both synthetic and real-world datasets.

Appendix

A Gaussian Loss with Different Divergence

In this paper, Gaussian Consistency loss involves three different forms to reduce mode collapse:

1) p-norm for the difference of mean and variance. To evaluate the divergence of two Gaussian distributions $\mathcal{N}(\mathbf{z}; \mathbf{0}, \boldsymbol{I})$ and $\mathcal{N}(\mathbf{z}; \tilde{\boldsymbol{\mu}}, \tilde{\mathbf{\Sigma}})$, we first calculate the difference of the parameters of Gaussian with p-norm:

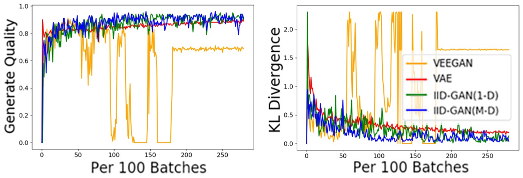
$$L_{Gau} = \|\tilde{\boldsymbol{\mu}}\|_p + \|\tilde{\mathbf{\Sigma}} - \mathbf{I}\|_p \tag{12}$$

2) Wasserstein distance. Given two M-D Gaussains $p(\mathbf{z})$ and $q(\mathbf{z})$, the 2-Wasserstein distance is: $W_2(p(\mathbf{z}), q(\mathbf{z})) = 0$ if and only if $\tilde{\mu} = 0$ and $\tilde{\mathbf{\Sigma}} = \mathbf{I}$.

3) KL divergence. Given M-D Gaussians $p(\mathbf{z})$ and $q(\mathbf{z})$, the KL divergence $KL(p(\mathbf{z}), q(\mathbf{z}))$ can be specified as:

$$L_{Gau} = \frac{1}{2}\left\{\log(\det(\tilde{\mathbf{\Sigma}})) - M + tr(\tilde{\mathbf{\Sigma}}^{-1}) + \tilde{\mu}^{\top}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mu}\right\} \tag{13}$$

Results of IID-GAN with different Gaussion losses evaluated on synthetic dataset are presented in Fig. 9.



(a) Generation quality (b) Inverse KL divergence

Figure 9: Comparison of Gaussian consistency loss on Ring data

B Experimental Details

B.1 Training Details for Synthetic Data

We set the loss weights as $(\lambda_{re}, \lambda_{Gau}) = (3, 10)$. Training batch size is set as 100 and we conduct 300 epochs for training. The metrics are calculated over 5 repetitions of evaluation.

B.2 Network Architectures for Real Data

The architectures for CIFAR-10 are presented in Table 7 and 8. The architectures for CELEBA and LSUN are presented in Table 9 and 10. The inverse mapping primarily follows the design of discriminator, with only the dimension of the output layer changed to match that of the latent space.

B.3 Training Details for Real Data

MNIST. We set the loss weights as $(\lambda_{re}, \lambda_{Gau}) = (0.5, 0.1)$. Following [Dieng *et al.*, 2019], we use the 10-category classifier to divide the generated images into 11 categories. If the highest probability of the generated image's prediction is smaller than 0.75, we treat it as a bad generation and classify it to the bad class. Otherwise, its label is determined according to the highest probability. Latent space dimension is set as 2. We train the models for 300 epochs.

StackedMNIST.. The StackedMNIST covers 1,000 known modes, as constructed by stacking three randomly sampled MNIST images along the RGB channels in line with the practice in [Metz *et al.*, 2017]. We follow [Srivastava *et al.*, 2017] to evaluate the number of covered modes and divergence between the real and generation distributions. The weights are set as $(\lambda_{re}, \lambda_{Gau}) = (3, 3)$. Latent space dimension is set as 100. We train the models for 50 epochs.

Layer	Output size
ConvTranspose2d	$2 \times 2 \times 1024$
BatchNorm2d	$2 \times 2 \times 1024$
Relu	$2 \times 2 \times 1024$
ConvTranspose2d	$4 \times 4 \times 512$
BatchNorm2d	$4 \times 4 \times 512$
Relu	$4 \times 4 \times 512$
ConvTranspose2d	$8 \times 8 \times 256$
BatchNorm2d	$8 \times 8 \times 256$
Relu	$8 \times 8 \times 256$
ConvTranspose2d	$16 \times 16 \times 128$
BatchNorm2d	$16 \times 16 \times 128$
Relu	$16 \times 16 \times 128$
ConvTranspose2d	$32 \times 32 \times 3$
Tanh	$32 \times 32 \times 3$

Table 7: Network Architectures of Generator G for CIFAR-10.

Layer	Output size
SN+Conv2d	$16 \times 16 \times 64$
LeakyRelu	$16 \times 16 \times 64$
SN+Conv2d	$8 \times 8 \times 128$
LeakyRelu	$8 \times 8 \times 128$
BatchNorm2d	$8 \times 8 \times 128$
SN+Conv2d	$4 \times 4 \times 256$
LeakyRelu	$4 \times 4 \times 256$
BatchNorm2d	$4 \times 4 \times 256$
SN+Conv2d	$2 \times 2 \times 512$
LeakyRelu	$2 \times 2 \times 512$
BatchNorm2d	$2 \times 2 \times 512$
faltten	2048
linear	1

Table 8: Network Architecture of Inverse D for CIFAR-10.

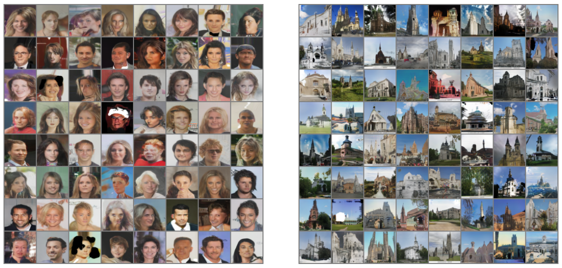
Layer	Output size
Linear+Reshape	$8\times8\times512$
ConvTranspose2d	$16 \times 16 \times 256$
BatchNorm2d	$16 \times 16 \times 256$
Relu	$16 \times 16 \times 256$
ConvTranspose2d	$32 \times 32 \times 128$
BatchNorm2d	$32 \times 32 \times 128$
Relu	$32 \times 32 \times 128$
ConvTranspose2d	$64 \times 64 \times 64$
BatchNorm2d	$64 \times 64 \times 64$
Relu	$64 \times 64 \times 64$
Conv2d	$64 \times 64 \times 3$
Tanh	$64 \times 64 \times 3$

Table 9: Architecture of G for CELEBA and LSUN.

Layer	Output size
SN+Conv2d	$64 \times 64 \times 64$
LeakyReLU	$64 \times 64 \times 64$
SN+Conv2d	$32 \times 32 \times 64$
LeakyReLU	$32 \times 32 \times 64$
SN+Conv2d	$32 \times 32 \times 128$
LeakyReLU	$32 \times 32 \times 128$
SN+Conv2d	$16 \times 16 \times 128$
LeakyReLU	$16 \times 16 \times 128$
SN+Conv2d	$16 \times 16 \times 256$
LeakyReLU	$16 \times 16 \times 256$
SN+Conv2d	$8 \times 8 \times 256$
LeakyReLU	$8 \times 8 \times 256$
SN+Conv2d	$8 \times 8 \times 512$
LeakyReLU	$8 \times 8 \times 512$
Faltten+Linear	1

Table 10: Architecture of D for CELEBA and LSUN.

CIFAR-10, STL-10, CELEBA and LSUN. All models are trained for 100K steps (mini-batches). We set $(\lambda_{re}, \lambda_{Gau}) = (0.01, 0.1)$. Latent space dimension is set as 128 and training batch size is set as 128. The metrics are calculated over 3 repetitions of evaluation. Fig. 10 presents the generation results of IID-GAN on CELEBA and LSUN Church datasets.



(a) CELEBA (b) LSUN Church

Figure 10: Generation results on CELEBA and LSUN Church for unconditional IID-GAN.

Contribution Statement

Yang Li and *Liangliang Shi* contribute equally to this work. *Junchi Yan* is the correspondence author. The work was partly supported by NSFC (62222607, U19B2035).

References

[Arjovsky *et al.*, 2017] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, pages 214–223, 2017.

[Bang and Shim, 2021] Duhyeon Bang and Hyunjung Shim. Mggan: Solving mode collapse using manifold-guided training. In *ICCV*, 2021.

[Coates *et al.*, 2011] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[De Meulemeester *et al.*, 2021] Hannes De Meulemeester, Joachim Schreurs, Michaël Fanuel, Bart De Moor, and Johan AK Suykens. The bures metric for generative adversarial networks. In *ECML PKDD*, 2021.

[Dieng *et al.*, 2019] Adji B Dieng, Francisco JR Ruiz, David M Blei, and Michalis K Titsias. Prescribed generative adversarial networks. *arXiv preprint arXiv:1910.04302*, 2019.

[Donahue *et al.*, 2017] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *ICLR*, 2017.

[Durugkar *et al.*, 2017] I. Durugkar, I. Gemp, and S. Mahadevan. Generative multi-adversarial networks. In *ICLR*, 2017.

[Elfeki *et al.*, 2019] Mohamed Elfeki, Camille Couprie, Morgane Riviére, and Mohamed Elhoseiny. Gdpp: Learning diverse generations using determinantal point processes. In *ICML*, 2019.

[Ghosh *et al.*, 2018] A. Ghosh, V. Kulharia, V. Namboodiri, P. H. Torr, and P. K. Dokania. Multi-agent diverse generative adversarial networks. In *CVPR*, 2018.

[Goodfellow *et al.*, 2014] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

[Gosset, 1987] Eric Gosset. A three-dimensional extended kolmogorov-smirnov test as a useful tool in astronomy. *Astronomy and Astrophysics*, 188:258–264, 1987.

[Grover *et al.*, 2018] Aditya Grover, Manik Dhar, and Stefano Ermon. Flow-gan: Combining maximum likelihood and adversarial learning in generative models. In *Proceedings of the AAAI conference on artificial intelligence*, 2018.

[Gulrajani *et al.*, 2017] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *NIPS*, 2017.

[Heusel *et al.*, 2017] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017.

[Higgins *et al.*, 2017] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Mohamed Shakir, and Alexander Lerchner. Learning basic visual concepts with a constrained variational framework. *ICLR*, 2017.

[Karras *et al.*, 2020] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*, 33:12104–12114, 2020.

[Kim and Mnih, 2018] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *ICML*, 2018.

[Kingma and Welling, 2014] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Tech Report*, 2009.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Li *et al.*, 2022] Yang Li, Yichuan Mo, Liangliang Shi, and Junchi Yan. Improving generative adversarial networks via adversarial learning in latent space. In *Advances in Neural Information Processing Systems*, 2022.

[Liu and Tuzel, 2016] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *NIPS*, 2016.

[Liu *et al.*, 2015] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[Liu *et al.*, 2020] Steven Liu, Tongzhou Wang, David Bau, Jun-Yan Zhu, and Antonio Torralba. Diverse image generation via self-conditioned gans. In *CVPR*, 2020.

[Mangalam and Garg, 2021] Karttikeya Mangalam and Rohin Garg. Overcoming mode collapse with adaptive multi adversarial training. *arXiv preprint arXiv:2112.14406*, 2021.

[Mao *et al.*, 2019] Qi Mao, Hsin-Ying Lee, Hung-Yu Tseng, Siwei Ma, and Ming-Hsuan Yang. Mode seeking generative adversarial networks for diverse image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[McCloskey and Cohen, 1989] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*. Elsevier, 1989.

[Metz *et al.*, 2017] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. In *ICLR*, 2017.

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

[Miyato *et al.*, 2018] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[Nguyen *et al.*, 2017] T. Nguyen, T. Le, H. Vu, and D. Phung. Dual discriminator generative adversarial nets. In *NIPS*, 2017.

[Peyré *et al.*, 2019] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[Radford *et al.*, 2016] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *ICLR*, 2016.

[Rezende and Mohamed, 2015] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *ICML*, 2015.

[Richardson and Weiss, 2018] Eitan Richardson and Yair Weiss. On gans and gmms. *arXiv preprint arXiv:1805.12462*, 2018.

[Salimans *et al.*, 2017] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *NIPS*, 2017.

[Shapiro and Wilk, 1965] Samuel Sanford Shapiro and Martin B Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611, 1965.

[Srivastava *et al.*, 2017] Akash Srivastava, Lazar Valkov, Chris Russell, Michael Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *NIPS*, 2017.

[Tran *et al.*, 2018] Ngoc-Trung Tran, Tuan-Anh Bui, and Ngai-Man Cheung. Dist-gan: An improved gan using distance constraints. In *ECCV*, 2018.

[Yang *et al.*, 2019] Dingdong Yang, Seunghoon Hong, Yunseok Jang, Tianchen Zhao, and Honglak Lee. Diversity-sensitive conditional generative adversarial networks. In *ICLR*, 2019.

[Yu *et al.*, 2015] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[Yu *et al.*, 2019] Xianwen Yu, Xiaoning Zhang, Yang Cao, and Min Xia. Vaegan: A collaborative filtering framework based on adversarial variational autoencoders. In *IJCAI*, pages 4206–4212, 2019.