IID-GAN: AN IID SAMPLING PERSPECTIVE FOR REGULARIZING MODE COLLAPSE

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite its success, generative adversarial networks (GANs) still suffer from mode collapse, namely the generator can only map latent variables to a partial set of modes of the target distribution. In this paper, we analyze and try to regularize this issue with an independent and identically distributed (IID) sampling perspective and emphasize that holding the IID property for generation for target distribution (i.e. real distribution) 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 from the target distribution. Based on this observation, we propose a necessary condition of IID generation and provide a new loss to encourage the closeness between the inverse source of real data and the Gaussian source in the latent space to regularize the generation to be IID from the target distribution. The logic is that the inverse samples from target data should also be IID in the source distribution. Experiments on both synthetic and real-world data show the effectiveness of our model.

1 INTRODUCTION

In the generative model fields, 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. Popular generative schemes include Variational Auto-Encoders (VAEs) (Kingma & Welling, 2014), Generative Flow models (Rezende & Mohamed, 2015), Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), among which GANs have been a popular tool for data generation, especially for generating images with a 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 excludes other parts of the target distribution (Liu et al., 2020; Yang et al., 2019), leading to a poor diversity of generation.

Efforts have been made to address mode collapse, along two branches: 1) get 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 may be the ultimate goal of generation, the two distributions are unknown, and we can only get the sampling data to improve the convergence. Thus the expression of the distribution convergence can be vague and imprecise. Besides, these methods study little the relation of the generated data, which leads them to improve convergence merely on the quality of the generated samples instead of diversity as the real data maintains. 2) penalize the similarity of the generated images (Elfeki et al., 2019; Meulemeester et al., 2020; Mao et al., 2019) or apply multiple generator/discriminators (Liu & Tuzel, 2016; Nguyen et al., 2017) . These methods increase the diverse generations directly but can hardly guarantee covering all the modes.

We first make a basic yet under-exploited (in GAN literature) observation that datasets for generation are assumed sampled to be independent and identically distributed (IID) from an unknown target distribution (i.e. real distribution). For the task of IID generation, **identical generation** from target distribution is the mainstream goal that previous works studied. For example, the discriminator D is used to distinguish whether every generation is sampled from an identical target distribution in GAN-based methods (Goodfellow et al., 2014). While in VAE-based or Flow-based approaches (Kingma & Welling, 2014; Rezende & Mohamed, 2015), Log-likelihood is applied to increase the density of every generation. Both of them focus on increasing the quality of generation but ignore the diversity.

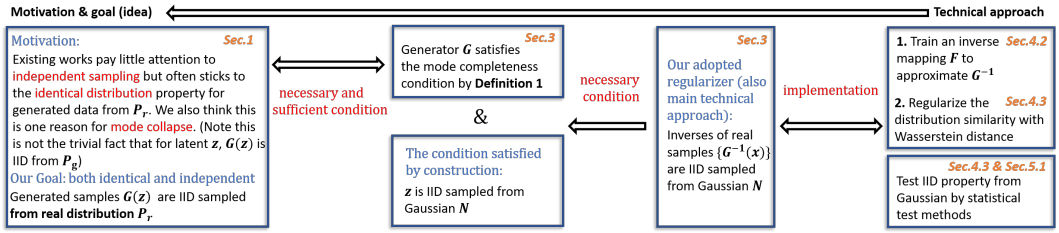


Figure 1: Main idea and technical logic. The left shows the motivation and ideal goal in Sec. 1. While the right part refers to our approach mainly in Sec. 4.

We aim to solve the problem of **independent generations** from the target distribution. Note that though the latent $\{z\}$ are IID sampled from the source, it cannot tell that the generations $\{G(z)\}$ are IID samples from the target distribution. We show more detailed idea as follows.

We first propose the concept of mode completeness as a core requirement of the 'perfect' mapping from the source distribution to the target distribution for IID generation. It is well known that the target distribution is unknown, and we only know its finite IID samples (i.e. real data). Thus we propose a weaker/necessary condition[1] for IID generation based on an inverse mapping perspective: if the real data are IID samples from target distribution, 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[2]. This idea also differs from existing inversion-based methods (Srivastava et al., 2017; Donahue et al., 2017; Rosca et al., 2017) which learn the relationship between the latent data $\mathbf{z}$ and real data $\mathbf{x}$ with the discriminator.

Our new perspective can also enrich and extend the understanding of mode collapse, whose concept has not been well established. Mode collapse in existing literature primarily refers to mode dropping. That is, the generation does not cover a few modes – see example in Fig. 2. Our IID view naturally allows for a flexible description of mode collapse, and the resulting new model is expected to handle both hard mode dropping and soft mode deficiency (i.e. some modes are not hit with enough generated samples as the real distribution). We show that the proposed method can be effective in both cases, as more appropriately measured by respective metrics. The main highlights of this paper are:

1) We take an IID perspective to address mode collapse in GAN. We first propose a perfect concept for generator, which should satisfy Mode Completeness as shown in Def. 1. Then due to the limitation of finite real samples from target distribution, we propose a necessary condition of mode completeness as shown in Prop. 1, which requires the IID property of the inverse samples from the real data.

2) With the requirement of Prop. 1, a regularizer is proposed for avoiding mode collapse. We enforce the inverse samples to be close to a standard Gaussian distribution by Wasserstein distance, which is termed as Gaussian consistency loss in this paper (see Section 4.3). QQ-plot, Shapiro–Wilk and Kolmogorov-Smirnov statistics are also 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.: number of covered modes, quality, reverse KL divergence. Our simple technique also performs competitively on natural images. We also investigate unsupervised disentanglement feature learning and conditional GAN. Our new regularization can be seen as an orthogonal plug-in to existing GAN models.

2 Related Work

Since its debut (Goodfellow et al., 2014), subsequent works of Generative Adversarial Networks (GANs) have been developed to improve the stability and quality of generation. Nevertheless GANs still suffer from training instability, and mode collapse has been one of the most common issues.

Improving training behavior. Unrolled GAN (Metz et al., 2017) presents a surrogate objective to train the generator, along with an unrolled optimization of the discriminator, which provides improvements in both training stability and distribution convergence. As an improvement of Wasserstein GANs (Arjovsky et al., 2017), WGAN-GP (Gulrajani et al., 2017) devises a gradient penalty. Although these methods all claim to improve the convergence between the generated and real distribu-

[1]Note that it will become a necessary and sufficient condition if real data is infinite.

[2]More precisely, in batch-based GAN training, each batch shall be close to standard Gaussian.

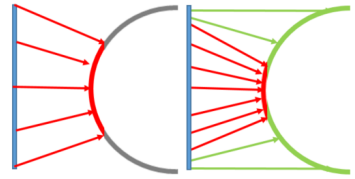
tion, the convergence may focus more on the quality of the generation, which makes every generated sample closer to the real distribution samples. However, they focus less on the relationship among the generated samples, making it hard to generate independent samples from the real distribution.

Enforcing to capture diverse modes. Many methods address GAN's mode collapse by increasing the diversity or penalizing similarity. GDPP (Elfeki et al., 2019) applies the determinantal point process theory, which gives a penalty for the discriminator to enforce the convergence of the covariance matrix between the features of generated samples and real data. The approach in (Meulemeester et al., 2020) uses the last layer output of the discriminator as a feature map to study the distribution of the real and the fake data. It matches the fake batch diversity with the real batch diversity by using the Bures distance between covariance matrices in feature space. All of these methods are concerned with the relationship among generated results through the covariance matrices. However, due to the randomness of generated samples and real samples, they lack theoretical guarantees and stability of the diversity generations, which do not capture the key point of generating data with the same diversity (i.e. the dependence of generations for real distribution).

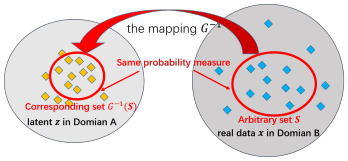
Multiple generators and discriminators. Another way to reduce mode collapse is involving more than one generator to achieve wider coverage for the real distribution. In (Liu & Tuzel, 2016), two coupled generator networks are trained with parameter sharing to learn the real distribution jointly. The multi-agent system MAD-GAN (Ghosh et al., 2018) involves multiple generators along with one discriminator. The system implicitly encourages each generator to learn its mode. On the other hand, multiple discriminators are used in (Durugkar et al., 2017) as an ensemble. Similarly, two additional discriminators are trained to improve the diversity (Nguyen et al., 2017). These models do not essentially solve mode collapse, and they resort more to network design and parameter tuning.

Mapping back to learn the representations. Methods e.g. BiGAN (Donahue et al., 2017), VEE-GAN (Srivastava et al., 2017) and VAE-based models (Kingma & Welling, 2014) design an inverse or encoding network of the generator to encourage the convergence between the inverse distribution of real data and source distribution. However, similar to the previously discussed methods which improve training behavior, the mapping back procedure also ignores the requirement of IID, which prevents convergence between the inverse and source distributions as shown in Fig. 4. Our work follows these works and takes one step further to show the effect of IID informed inverse mapping for solving the mode collapse issue.

These efforts are orthogonal to ours and most of them can be fulfilled in conjunction with ours to further improve the training stability, which we leave as future work.



(a) two mode collapse cases



(b) mode completeness

Figure 2: Mapping from source to target: (a) hard mode drop (left) and soft mode deficiency (right). Mode dropping can be treated as a special case of the soft mode deficiency when the green part is sparse to the limit. (b) Assume the existence of the inverse of mapping $G(\cdot)$. Given any set S, the probability measures of the set $G^{-1}(S)$ and S are equal.

3 Preliminaries and Motivation

Motivated by the mode collapse case presented in Fig. 2(a), we define the mode completeness as the ideal mapping state between two probability measures for the generator:

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.

Though the purpose of mode completeness is to avoid mode collapse as shown in Fig. 2(b), Eq. 1 is exactly the same as the definition of the push-forward operator (Peyré et al., 2019) $\beta = G_{\#}\alpha$ in optimal transportation. So the mode completeness can also be understood as designing the mapping G that satisfies that $\beta = G_{\#}\alpha$ and the operator $G_{\#}$ means that G pushes forward the mass of α to β (Peyré et al., 2019). Based on Def. 1, we can find that the mode collapse will not happen because of the same value of the probability measures given the corresponding sets based on G (i.e. $\alpha(\{\mathbf{z}_i\}) = \beta(\{T(\mathbf{z}_i)\})$). Then when $\mathbf{z}_i$ are IID

sampled from α, the generated samples $\{G(\mathbf{z}_i)\}$ can also be viewed as IID from β due to the equal probability.

The IID generation, i.e. mode completeness $\beta = G_{\#}\alpha$ can not completely achieve for the generative model because β is complex and unknown. The only information that we will make full use of, is the IID basic assumptions of the real training data from β. If we assume the existence of the inverse of the mapping G i.e. $G^{-1} : \mathcal{B} \to \mathcal{A}$, which satisfies $z = G^{-1}(G(z))$ 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 to address the mode collapse. Such a necessary condition for IID generation is formalized by the following proposition (see Appendix A for proof):

Proposition 1 (IID Property for Inverse Targets) *If the generative mapping G satisfies mode completeness from the source probability measure α to the target probability 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.*

Remarks: Since it is commonly assumed that the real data $\{\mathbf{x}^{(i)}\}_{i=1}^n$ are independently sampled from an unknown distribution β, we can obtain that their inverses $\{G^{-1}(\mathbf{x}^{(i)})\}$ can be viewed as IID samples from a known distribution α. So to satisfy the mode completeness, we need to train the inverse mapping G^{-1} so that the inverse samples of real data can be closer to the IID samples from α. Note we cannot obtain a strict inverse mapping G^{-1}, a function F which maps back to the source space will be designed to approximate G^{-1} by penalizing $\mathbf{z} = F(G(\mathbf{z}))$ and $x = G(F(\mathbf{x}))$.

Essential difference to VEEGAN. VEEGAN presents a similar idea with Prop. 1, which maps back and enforces the inverse samples to obey Gaussian by discriminating real/fake $(\mathbf{z}, \mathbf{x})$ pair. However, the essential difference is that VEEGAN do **not** take an **IID perspective**. It is well known that the domain of (1D) Gaussian distribution is over R and thus every inverse sample can naturally be viewed as a Gaussian sample. What the discriminator of VEEGAN actually does is to enforce $G^{-1}(\mathbf{x})$ to be close to 0. Differently, Prop. 1 requires the entire inverses $\{G^{-1}(\mathbf{x})\}$ to be IID sampled from Gaussian, which require us to consider the inverse sample set together. Fig. 4 shows the failure of VEEGAN to approximate the inverse samples to Gaussian.

4 The Proposed Approach

Our loss (including conditional GAN (Mirza & Osindero, 2014)) contains three parts – see Fig. 3.

4.1 Adversarial Learning Term: Vanilla GAN Loss

The vanilla GAN model (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 $p(x)$, $D(\mathbf{x})$ is used to distinguish whether the generated samples from real data, while $G(\mathbf{z})$ is the mapping from Gaussian sample $\mathbf{z}$ to a point in the target space $\mathcal{R}^d$. The objective $V(G, D)$ is optimized for the discriminator and generator by alternatingly solving the mini-max:

$$E_{\mathbf{x}\sim p_r(\mathbf{x})}\left[\log(D(\mathbf{x}))\right] + E_{\mathbf{z}\sim p(\mathbf{z})}\left[\log(1 - D(G(\mathbf{z})))\right] \tag{2}$$

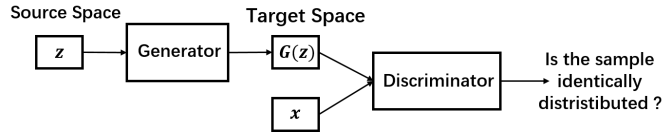
The first term denotes the probability expectation that $\mathbf{x}$ comes from real data distribution $p(x)$ and the second involves the input distribution $p(\mathbf{z})$, which is embodied in this paper as a standard multi-dimensional (M-D) Gaussian distribution $\mathcal{N}(\mathbf{z}; \mathbf{0},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

4.2 Reconstruction Term: Cycle-Consistency Loss for Single Sample

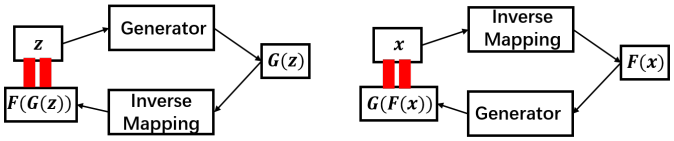
Prop. 1 requires an inverse mapping G^{-1} of the generation. Noting that our method involves inverse mapping as implemented by neural network F, a popular and effective technique (Srivastava et al., 2017; Donahue et al., 2017) is to employ a cycle-consistency loss to prompt $F = G^{-1}$, where $\lambda = d/M$ is the dimensionality ratio of $\mathbf{x}$ to $\mathbf{z}$.

$$L_{re}(G, F) = \underbrace{E_{\mathbf{z}}\|\mathbf{z} - F(G(\mathbf{z}))\|_1}_{\text{to achieve inverse mapping}} + \lambda \underbrace{E_x\|\mathbf{x} - G(F(\mathbf{x}))\|_1}_{\text{to avoid mode dropping}} \tag{3}$$

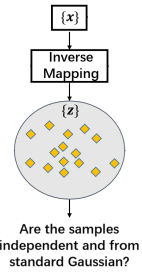
The first term promotes F to be the inverse of G, which takes the reconstruction loss as the expectation of the cost of auto-encoding noise vectors (Srivastava et al., 2017). The second term promotes $F(\mathbf{x}) \in \mathcal{R}^M$, which makes $\tilde{\mathbf{z}} = F(\mathbf{x})$ possible to be sampled from Gaussian distribution. Then for each $\mathbf{x}$, we can find the corresponding $\tilde{\mathbf{z}}$ in $\mathcal{R}^M$ satisfying $G(\tilde{\mathbf{z}}) = \mathbf{x}$. Note that it cannot help avoid mode imbalance, since the imbalance refers to the overall distribution rather than the individual data points studied here. Then we introduce our source space distribution closeness loss.



(a) Loss to guarantee identical sampling



(b) Cycle-Consistency for inverse mapping



(c) Loss to guarantee independent sampling

Figure 3: IID-GAN: generator G maps random samples from original standard M-D Gaussian to target ones and F inverts the target sample back to a source sample which obeys M-D Gaussian. Note that CycleGAN addresses image-to-image generation and studies the mapping between image domains, while we only study the mapping relationship between latent space and target space.

4.3 Regularizer Term: Gaussian Consistency Loss for Inverse Distribution

Recalling the necessary condition for IID generation proposed in Prop. 1, assume that the given real data $\{\mathbf{x}^{(i)}\}$ 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, the mapping F should make $\{F(\mathbf{x}^{(i)})\}$ closer to independent samples from standard Gaussian $p(\mathbf{z})$. For simplicity, we choose Gaussian as the distribution for IID sampling.

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

$$\tilde{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N}\tilde{\mathbf{z}}^{(i)}, \quad \tilde{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{i=1}^{N}\left(\tilde{\mathbf{z}}^{(i)} - \tilde{\boldsymbol{\mu}}\right)^{\top}\left(\tilde{\mathbf{z}}^{(i)} - \tilde{\boldsymbol{\mu}}\right) \tag{4}$$

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 $p(\mathbf{z})$. One way is to introduce a distribution closeness loss L_{Gau}. E.g., it can be specified as the square of Wasserstein distance of two Gaussians:

$$L_{Gau} = \|\tilde{\boldsymbol{\mu}}\|^2 + trace(\tilde{\boldsymbol{\Sigma}} + \mathbf{I} - 2\tilde{\boldsymbol{\Sigma}}^{1/2}) \tag{5}$$

The two Gaussian distributions $q(\mathbf{z})$ and $p(\mathbf{z})$ can also be evaluated with static divergence or directly distinguished by a designed discriminator, e.g., p-norm, KL-divergence or $\tilde{\mathbf{z}}-$discriminator. Note that the method (Makhzani et al., 2015) with $\tilde{\mathbf{z}}-$discriminator trains another discriminator to distinguish the Gaussian samples. More details about these methods are presented in Appendix B.

Decoupling M-D Gaussian into M 1-D Gaussians. For large M and small batch size, the training can suffer from the curse of dimensionality for estimating $\tilde{\boldsymbol{\Sigma}}$. So we decouple the M-D Gaussian loss to the sum of the 1-D Gaussian losses of M ones. Specifically, we introduce 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 $\tilde{\boldsymbol{\sigma}}$ for the Gaussians.

Here our goal is to make the 1-D Gaussian distribution $q(\mathbf{z}_j)$ more 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 them over M dimensions:

$$L_{Gau} = \sum_{m=1}^{M}\left(\tilde{\boldsymbol{\mu}}_m^2 + (\tilde{\boldsymbol{\sigma}}_m - 1)^2\right) \tag{6}$$

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

The final objective. By inputting as many samples sampled from $p(\mathbf{z})$ as possible, together with the corresponding inverse samples $\{\tilde{\mathbf{z}}\}$ from the real data, we optimize D_z 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:

$$\min_{G,F}\max_{D} V(G, D) + \lambda_{re}L_{re}(G, F) + \lambda_{Gau}L_{Gau}(F) \tag{7}$$

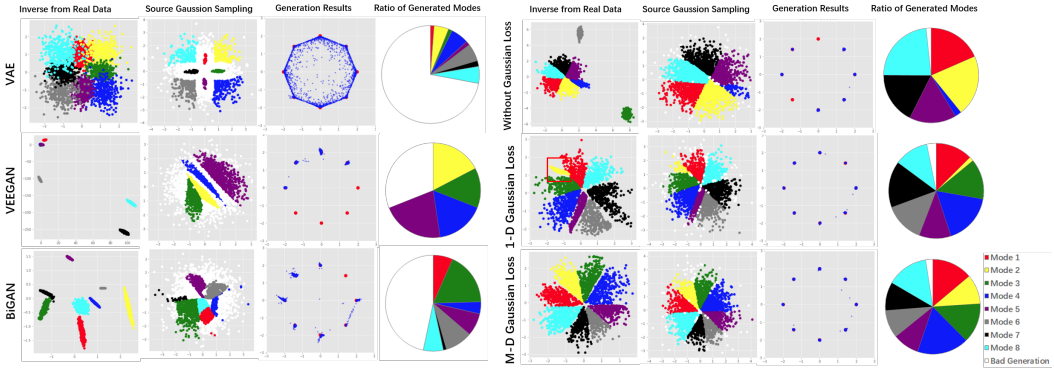


Figure 4: Example for 2-D source ($\mathbf{z}$) to 2-D target ($\mathbf{x}$) generation and inverse with 8 modes Ring dataset as the real training set. Left 4 columns are the results of VAE, BiGAN, VEEGAN, while right 4 columns are the results of IID-GAN under different Gaussian consistency losses as 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. Points in source 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.

where $L_{Gau}(F)$ can be specified according to different distances or divergence and $\lambda_{re}, \lambda_{Gau}$ are loss weights, which will be discussed in experiment in detail.

IID-GAN can be extended to the conditioned case (Mirza & 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 for generating data points and the real $\mathbf{x}$ is the input of F for classification, and the Gaussian consistency loss is used to maintain the independent condition. The Gaussian loss is mainly used to learn the diversity of hidden features (e.g. the thickness, inclination for MNIST). See details in Appendix C.

Remarks for the Disentanglement View. Many previous studies (Higgins et al., 2017; Kim & Mnih, 2018) learn the unsupervised disentanglement representation with the assumption of independent factors, i.e. $q(\mathbf{z}) = \prod_{j=1}^{M} q(\mathbf{z}_j)$. However, the recent work (Locatello et al., 2019) opposes this view and argues that unsupervised learning of disentangled representations is not possible without inductive biases on both models and data. 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 first column of Fig. 4. We find that representing the data in polar coordinates and varying the polar angle and diameter may be a good way for unsupervised learning to disentangle representations.

IID testing for Gaussian. In this subsection, we aim to guarantee the IID Gaussian property for inverse samples from the real data. However, the regularization assumes a non-standard Gaussian and enforces the consistency of two Gaussians, which raises doubts: 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), Kolmogorov-Smirnov test (KS) to show whether the samples are IID sampled from standard Gaussian. The results are shown in Fig. 6 and Table 2.

4.4 Further Comparisons with Peer Approaches

In this subsection, we fully discuss the comparisons with peer approaches in terms of methodology.

Comparison to VAE-based methods. Most VAE-based methods (Rosca et al., 2017; Higgins et al., 2017; Kim & Mnih, 2018) contain an encoder and use the Gaussian distributions for each real data to learn the latent representation. The difference is that our method considers the overall mini-batch data to make the inverse samples of the real data approximate IID standard Gaussian samples, based on which our method can generate more accurate and diverse data.

Comparison to GAN methods with inverse mapping. Many GAN based methods (Donahue et al., 2017; Jeff & Simonyan, 2019) also try to learn the representation by inverse mapping. However, in contrast to IID-GAN, these methods do not consider the data from the perspective of the overall samples and do not consider the IID property. CycleGAN (Zhu et al., 2017) proposes a cycle consistency loss to obtain the transition between two different styles of images. However, different

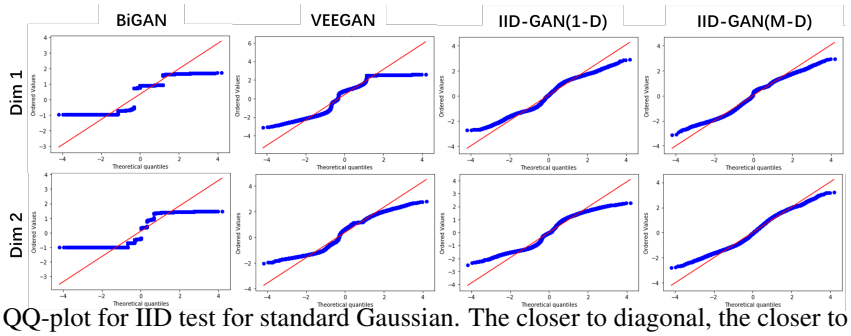


Figure 6: QQ-plot for IID test for standard Gaussian. The closer to diagonal, the closer to Gaussian.

from our method, CycleGAN is designated for image translation and pays less attention to the diversity of the generated images. Their method has nothing to do with IID.

Comparison of M-D and 1-D Gaussian loss. Our method claims the Gaussian necessary conditions for the inverse samples $\{\tilde{\mathbf{z}}\}$ of real data. $\{\tilde{\mathbf{z}}\}$ is a multi-dimensional vector in latent space and we expect the inverse samples to obey the standard Gaussian $\mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$, which is the goal of M-D Gaussian loss. However, in the case of training with real-world data, $\{\tilde{\mathbf{z}}\}$ may have a high dimension and lead to a dimension curse with small batch size. We consider estimating M single-dimensional standard Gaussian along each dimension in the latent space to address this problem, as shown by the delegated loss in Eq. 6.

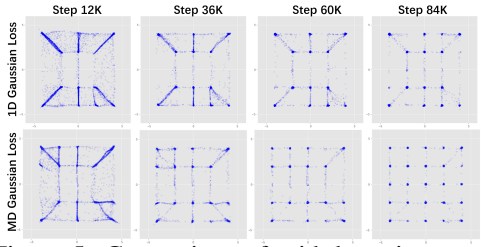


Figure 5: Generations of grid data given poor initialization. Compared with 1-D Gaussian consistency loss, the M-D loss outperforms after enough batch iterations (i.e. training steps). Similar results are shown on Ring in Appendix F.

5 Experiments and Discussion

In this section, we adopt a simpler network architecture to directly evaluate the superiority of our techniques. All experiments are conducted on a single GPU of GeForce RTX 2080Ti.

5.1 Experiments on Synthetic Datasets

Since the distribution is known, mode collapse can be directly measured on synthetic data. In line with (Metz et al., 2017), we simulate two synthetic datasets.

Ring dataset. The dataset 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 (i.e. 100K samples in total). 50K samples from $p(\mathbf{z})$ are used to generate $\mathbf{x}$ for test.

Grid dataset. The dataset 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 (i.e. 100K samples in total). 100K samples from $p(\mathbf{z})$ are used to generate target samples $\{\tilde{\mathbf{x}}\}$ for test.

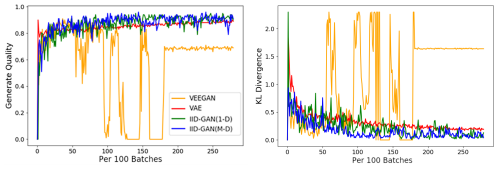
Metrics and network architecture. Following (Metz et al., 2017; Elfeki et al., 2019), we use the number of covered modes, generation quality[3] and reverse KL divergence. Since in the experiment, each mode shares the same number of real samples, one can calculate the reverse KL divergence between the generated distribution and the real one (Nguyen et al., 2017) . The reverse KL divergence is not strictly defined as $\sum_{i=1}^{m} p_i < 1$ (i.e. there exist poor generated points), and allows being negative. And for the network architecture, we adopt ReLU instead of sigmoid as the activation function and the networks consist of four linear layers. The architecture details are in Appendix D .

Results with Generation Metrics. IID-GAN is compared with 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. We can see that the two IID-GAN cover all modes on both Ring and Grid. And for the RKL evaluation, IID-GAN(M-D) achieves a lower RKL value and a smaller standard deviation. Besides, the quality metric of IID-GAN remains high, which indicates that the inverse mapping has no bad effect during training on synthetic datasets.

[3]We follow (Meulemeester et al., 2020): if the generated data point is within 3 times the std of the Gaussian, consider it a good (or valid) generation (otherwise bad) and the resulting ratio is used as the generation quality.

Table 1: The generation results for Ring and Grid synthetic data.

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	$\mathbf{98.0 \pm 0.4}$	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)	$\mathbf{8.0 \pm 0.0}$	$\mathbf{99.0 \pm 0.2}$	$\mathbf{0.17 \pm 0.06}$	$\mathbf{25.0 \pm 0.0}$	$\mathbf{98.0 \pm 0.4}$	$\mathbf{0.26 \pm 0.12}$



(a) Generation Quality (b) KL divergence

Figure 7: Generating quality and KL divergence (for diversity) from the inverse source to the standard Gaussian on MNIST.

Table 3: Results on CIFAR-10 and CIFAR-100.

Models	CIFAR-10		CIFAR-100	
	IS↑	FID↓	IS↑	FID↓
GAN	4.84	78.4	4.79	85.6
Unrolled GAN	4.65	76.2	4.96	83.1
VEEGAN	3.56	161.1	4.34	88.6
GDPP	4.43	80.4	4.87	82.8
DP-GAN	4.72	78.7	4.79	83.3
IID-GAN(1-D)	4.89	**65.4**	5.05	84.5
IID-GAN(M-D)	**5.00**	76.9	**5.27**	**82.1**

The Gaussian inverse samples. As discussed above, some methods (Kingma & Welling, 2014; Srivastava et al., 2017) designs an inverse mapping or an encoder to learn the representations between $\mathbf{z}$ and $\mathbf{x}$. However, as shown in Left 4 columns of Fig. 4, VAE-based methods (Kingma & Welling, 2014) can cause the overlap of the inverse samples $\mathbf{z}$, which may lead to bad generations(white points). And BiGAN and VEEGAN learn the relations between $\mathbf{z}$ and $\mathbf{x}$ rather than the relation among the samples $\{G^{-1}(\mathbf{x})\}$, which fails to get the 2D inverse Gaussian samples as shown in the first column of Fig. 4. For IID-GAN, as shown in the fifth column for IID-GAN(M-D), the inverse samples is very similar to the Gaussian samples, which show the effect of the regularization.

Generation with bad initialization. It is well known that bad initializations can usually lead to the mode collapse. However, as shown in Fig. 5, IID-GAN(M-D) can overcome the problem of bad initialization and gradually solve the mode collapse by further training.

IID Test for inverse samples. As discussed in Sec. 4.3, we conduct the IID test for Gaussian distribution on Ring dataset. Shapiro-Wilk(SW) and Kolmogorov-Smirnov(KS) Statics are calculated to show the IID property. We present the results in Table 2. Note that SW static is up to 1, while KS Statics is down to 0. Besides, in Fig. 6, QQ-plot are used for IID test. Compared to other methods, the blue points of IID-GAN are closer to the red diagonal, which means that the inverse samples of each dimension are close to the Gaussian.

Table 2: Static evaluation for IID test on Ring.

Model	SW Static ↑	KS Static ↓	
		Dimension 1	Dimension 2
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**

Ablation study for Gaussian consistency loss. On the right four columns of Fig. 4, we compare the results of 1-D, M-D and without Gaussian consistency loss in Ring datasets and notice that the M-D Gaussian setting perform better. Specifically, for such a 2-D source space case, the yellow inverse samples in red boxes lead to soft mode deficiency in 1-D Wasserstein loss setting, while the generated modes in the M-D case are more uniformly scaled and more suitable to solve the soft mode deficiency. More results about synthetic data are shown in Appendix F.

5.2 Experiments on Real-world Data

The experimented image datasets include MNIST, stackedMNIST, CIFAR-10, CIFAR-100, and STL-10. For all experiments, the model is trained for 300 epochs with a batch size of 256 and 0.0002 learning rate. We adopt different architectures to evaluate our model, which mainly follow the previous studies (Radford et al., 2016; Meulemeester et al., 2020; Dieng et al., 2019).

All the compared models are trained in 100K steps and the results are calculated based on 10K generated images for CIFAR-10, CIFAR-100 and STL-10 (Coates et al., 2011). Detailed information about the network architectures for experiments is presented in Appendix G.1. Here we use p-norm distance as M-D Gaussian loss. It is worth noting that our IID-GAN rarely encounters training failures e.g. gradient explosion which most other compared methods struggle during training. Besides, for the evaluation, we adopt the popular Inception Score (IS) (Salimans et al., 2017) and Fréchet

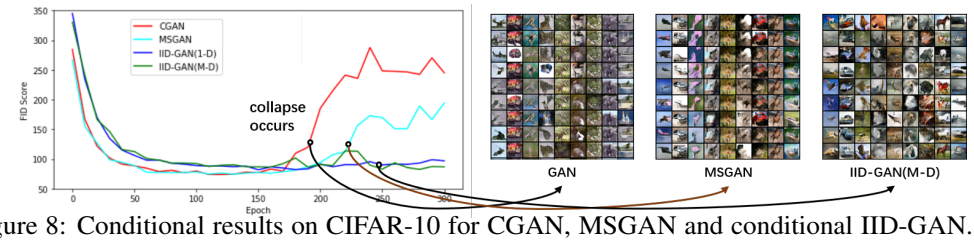


Figure 8: Conditional results on CIFAR-10 for CGAN, MSGAN and conditional IID-GAN. The generation results of CGAN and MSGAN can easily deteriorate as training proceeds, while IID-GAN maintains good performance consistently.

Inception Distance (FID) (Heusel et al., 2017) as quantitative metrics. Mode Score (MS) (Che et al., 2017) is adopted to make full use of information from real datasets for reasonable evaluation. Following Richardson & Weiss (2018), JSD is used to measure the similarity between the generated distribution and the real distribution. For easily distinguishable datasets like MNIST, we use the number of covered modes and reverse KL for evaluation. The training details are in Appendix E .

Training stability for IID-GAN. As shown in Fig. 7, the generation quality and KL divergence are evaluated on MNIST. 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.

Evaluation Results. In addition to MNIST, the results of stacked MNIST are shown in Table 4, which is evaluated by Mode Number, KL divergence and FID. The KL divergence calculation is based on the classification results by the classifier proposed by (Dieng et al., 2019). Table 5 shows the experimental results of CIFAR-10 and CIFAR-100. We evaluate the generations of IID-GAN with IS and FID, comparing with UnrolledGAN (Metz et al., 2017), VEEGAN (Srivastava et al., 2017), GDPP and DP-GAN (Pei et al., 2021). The generation of STL-10 is evaluated in Table 5. These experimental results exhibit the superiority of IID-GAN.

Table 4: Generation Results on Stacked MNIST and the architecture is in line with Radford et al. (2016).

Models	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 5: Generation Results on STL-10.

Models	STL-10		
	IS↑	FID↓	MS↑
GAN	2.28	245.21	2.29
BiGAN	1.22	251.21	1.22
Unrolled GAN	4.78	142.16	4.62
VEEGAN	1.45	298.95	1.46
IID-GAN(M-D)	**5.16**	**139.10**	**5.12**

Results on conditional generation. We present the results in Fig. 8. Given different categories for generation, the conditional IID-GANs (including its 1-D version) are the most stable and robust on CIFAR-10, and do not suffer from mode collapse. More results about conditional generation are presented in Appendix G.2.

Table 6: Evaluation with the framework of WGAN-GP and SNGAN. .

Models	CIFAR-10	
	IS↑	JSD↓
WGAN-GP	7.343	0.00339
IID-GAN(WGAN-GP)	**7.443**	**0.00326**
SNGAN	7.255	0.0327
IID-GAN(SNGAN)	**7.350**	**0.0219**

Results on different GAN frameworks. The previous results are based on vanilla GAN. To fit with state-of-the-art generative model achievements, we introduce IID reorganizations on advanced models e.g. WGANGP (Gulrajani et al., 2017) and SNGAN (Miyato et al., 2018), to examine the performance of IID reorganizations on more frameworks. We conducted experiments on CIFAR-10. As shown in Table 6, the performance of WGAN-GP and SN-GAN is improved under the IID regularization.

Results on disentanglement. Unsupervised disentanglement results are shown in Fig. 12 in Appendix. We study it with the polar coordinate system instead of Cartesian coordinates. By varying the polar radius and the polar angle, we obtain a good disentanglement result. As shown in Fig. 11, we can see the disentanglement results in different coordinate systems. More results about real data are shown in Appendix G.2.

6 Conclusion

We provide an IID sampling perspective to address the mode collapse of GAN. Our devised inverse mapping technique and the new loss show their effectiveness in solving mode collapse on both synthetic and real-world datasets. The source code will be made publicly available.

REFERENCES

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, pp. 214–223, 2017.

Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode regularized generative adversarial networks. In *ICLR*, 2017.

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*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

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

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

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

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

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

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.

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

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.

Donahue Jeff and Karen Simonyan. Large scale adversarial representation learning. *arXiv preprint*, 2019.

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

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

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.

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

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

Francesco Locatello, Mario Bauer, Stefan andLucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *ICLM*, 2019.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *ICML*, 2015.

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.

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

Hannes Meulemeester, Joachim Schreurs, Michael Fanuel, Bart Moor, and Johan Suykens. The bures metric for taming mode collapse in generative adversarial networks. In *CVPR*, 2020.

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

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

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

Sen Pei, Richard Yi Da Xu, and Gaofeng Meng. dp-gan: Alleviating mode collapse in gan via diversity penalty module. *arXiv preprint arXiv:2108.02353*, 2021.

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.

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

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

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

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

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

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

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

APPENDIX

A DETAILS IN SECTION 3

A.1 A GENERAL CASE OF PROPOSITION 1

Proposition 2 *Assume existence of the mapping G which satisfies that $G_{\#}\alpha = \beta$ and its inverse G^{-1}. Let ρ is the joint probability measure of n independent β and π is the joint probability measure of n α where $\tilde{T}_{\#}\pi = \rho$ and $\tilde{T} = [T, T, \ldots, T]$ with concat of n mapping T. then we can get that π is the joint probability measure with n **independent** α.*

Proof 1 *Given n measurable set $\mathcal{S}_i \subset \mathcal{B}$ and $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \times \mathcal{S}_n$, then we can get that*

$$\rho(\mathcal{S}) = \beta(\mathcal{S}_1)\beta(\mathcal{S}_2)\ldots\beta(\mathcal{S}_n) \tag{8}$$

For the reason that $\tilde{T}_{\#}\pi = \rho$ and $T_{\#}\alpha = \beta$, we know that $\rho(\mathcal{S}) = \pi(\tilde{T}^{-1}(\mathcal{S}))$ and $\beta(\mathcal{S}_i) = \alpha(T^{-1}(\mathcal{S}_i))$. And then we can get that

$$\pi(\tilde{T}^{-1}(\mathcal{S})) = \alpha(T^{-1}(\mathcal{S}_1)) \cdot \alpha(T^{-1}(\mathcal{S}_2))\ldots\alpha(T^{-1}(\mathcal{S}_n)) \tag{9}$$

*which means that π is the joint probability measure with n **independent** α.*

A.2 THE PROOF OF PROPOSITION 1

Proposition 1 is a special case of Proposition 2. So the proof of Proposition 1 can easily get from the proof of Proposition 2.

Proof 2 *Set n measurable sets as $\mathcal{S}_1 = \{x_1\}, \mathcal{S}_2 = \{x_2\}, \ldots, \mathcal{S}_n = \{x_n\}$ where $x_1, x_2, \ldots, x_n$ are n IID samples from target distribution, then according to Proposition 2, we can get that the following equation with Eq. 9:*

$$\alpha(T^{-1}(x_1)) \cdot \alpha(T^{-1}(x_2))\ldots\alpha(T^{-1}(x_n)) = \pi(\tilde{T}^{-1}(\{x_1, \ldots, x_n\})) \tag{10}$$

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

B MORE M-D GAUSSIAN LOSS WITH DIFFERENT DIVERGENCE

In this paper, four methods are used to optimize as Gaussian Consistency loss to reduce the 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 $\mathcal{N}(\mathbf{z}; \tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}})$, we first calculate the difference of the parameters of Gaussian with p-norm:

$$L_{Gau} = \|\tilde{\boldsymbol{\mu}}\|_p + \|\tilde{\boldsymbol{\Sigma}} - \mathbf{I}\|_p \tag{11}$$

2) Wasserstein distance. The Wasserstein distance has been widely used to evaluate the distance between two distributions. Given two M-D Gaussains $p(\mathbf{z})$ and $q(\mathbf{z})$, the 2-Wasserstein distance is: We can see that $W_2(p(\mathbf{z}), q(\mathbf{z})) = 0$ if and only if $\tilde{\mu} = 0$ and $\tilde{\boldsymbol{\Sigma}} = \mathbf{I}$.

3) KL divergence. It is an important divergence to measure the difference between two distributions. 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:

$$L_{Gau} = \frac{1}{2}\left\{\log(\det(\tilde{\boldsymbol{\Sigma}})) - M + tr(\tilde{\boldsymbol{\Sigma}}^{-1}) + \tilde{\mu}^\top \tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\mu}\right\} \tag{12}$$

4) The $\tilde{\mathbf{z}}-$discriminator. Real discriminators distinguish whether the generated image is a sample of real distribution $p(x)$. Similarly, we can also use a discriminator to distinguish the difference between real Gaussian samples and generated ones as is done in Makhzani et al. (2015). We introduce a discriminator to distinguish whether it is from standard Gaussian distribution. We can get the final loss as

$$\min_{G,F}\max_{D,D_z} V(G, D) + L_{cons}(G, F) + L_{Gau}(F, D_z) \tag{13}$$

where $L_{Gau}(F, D_z)$ can be defined as

$$E_{\mathbf{z}\sim P_z}\left[\log(D_z(\mathbf{z}))\right] + E_{x\sim p(x)}\left[\log(1 - D_z(F(x)))\right] \tag{14}$$

Through alternating training, we can get the optimal G, F and D, D_z.

C CONDITIONED IID-GAN

When the conditioned label c is known, the objective function $V(G, D)$ is optimized for the discriminator and generator by solving the minimax problem in an alternating fashion as:

$$E_{x\sim p(x)}\left[\log(D(x))\right] + E_{\mathbf{z},\mathbf{c}\sim p(\mathbf{z})}\left[\log(1 - D(G(\mathbf{z},\mathbf{c})))\right] \tag{15}$$

The first term gives the expectation of probability that x comes from real data distribution $p(x)$ and the second involves an input distribution $p(\mathbf{z}, \mathbf{c})$, which is embodied by a standard multi-dimensional(M-D) Gaussian distribution $\mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$ and discrete uniform distribution in this paper. To get the approximate inverse mapping, We adopt the neural network F as the inverse of the generator G. The reconstruction loss is specified as:

$$L_{re}(G, F) = E_{\mathbf{z},\mathbf{c}}\| \begin{bmatrix}\mathbf{z}\\ \mathbf{c}\end{bmatrix} - F(G(\mathbf{z},\mathbf{c}))\|_2 + E_x\|x - G(F(x))\|_2 \tag{16}$$

where $\mathbf{z}, \mathbf{c}$ are the inputs of the generator to generate the data points x and given x, the inverse mapping F will reconstruct source sample z and label c. Besides, Gaussian loss for conditional IID-GAN are similar to unconditioned case.

D SYNTHETIC DATA

D.1 NETWORK ARCHITECTURES FOR SYNTHETIC DATA

Instead of using tanh as the activation function as adopted in Metz et al. (2017); Elfeki et al. (2019) for more stable training, to more directly verify our technique, we resort to ReLU with four linear layers as the network architecture.

Table 7: Network Architecture of Inverse F for Synthetic Ring-Grid Data.

Layer	Output size	Activation
Linear	100	ReLu
Linear	200	ReLu
Linear	100	ReLu
Linear	2	-

Table 8: Network Architecture of Discriminator D for Synthetic Ring-Grid Data.

Layer	Output size	Activation
Linear	100	ReLu
Linear	200	ReLu
Linear	100	ReLu
Linear	2	-

Table 9: Network Architecture of Discriminator D for Synthetic Ring-Grid Data.

Layer	Output size	Activation
Linear	100	ReLu
Linear	200	ReLu
Linear	100	ReLu
Linear	1	-

E TRAINING DETAILS

MNIST MNIST contains 70,000 images of handwritten digits (LeCun et al., 1998). KL divergence is used for evaluation. We set the weights $(\lambda_{re}, \lambda_{Gau}) = (0.5, 0.1)$. Following Dieng et al. (2019), a classifier is trained to distinguish the category of generation images. We use the 10-category classifier to divide the generated images into 11 categories. If the highest probability of the generated picture's prediction is smaller than 0.75, it means that the generation quality is poor and classified as bad, otherwise, its label is determined according to the highest probability.

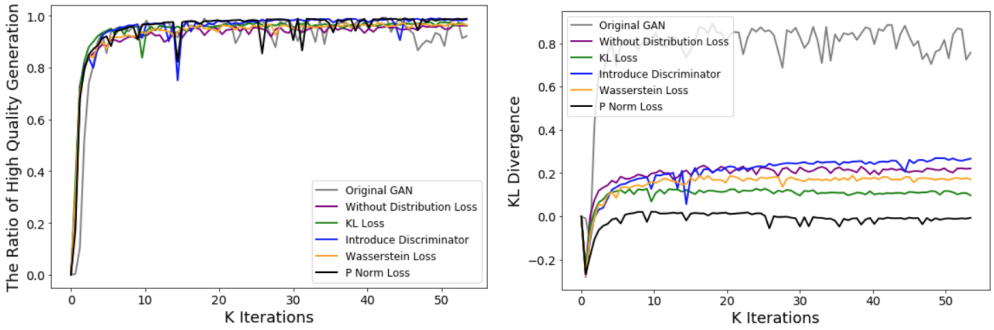
StackedMNIST. Our StackedMNIST covers 1,000 known modes, as constructed by stacking three randomly sampled MNIST images along the RGB channels in line with the practice in Srivastava et al. (2017). We also follow Srivastava et al. (2017) to evaluate the number of covered modes and divergence between the real and generation distributions. The weights are set $(\lambda_{re}, \lambda_{Gau}) = (3, 3)$. **Results on CIFAR-10 and CIFAR-100.** All models are trained for 100K steps i.e. mini-batches.

We set $(\lambda_{re}, \lambda_{Gau}) = (3, 3)$. In Table 5, Inception Score and Fréchet Inception Distance (FID) are used to evaluate the image quality and mode completeness. The best performance is observed for IID-GAN w.r.t. image quality, measured by FID and Inception score and mode completeness, measured by FID. Table 3 shows that IID-GAN outperforms UnrolledGAN, VEEGAN and GDPP.

F SYNTHETIC RESULTS

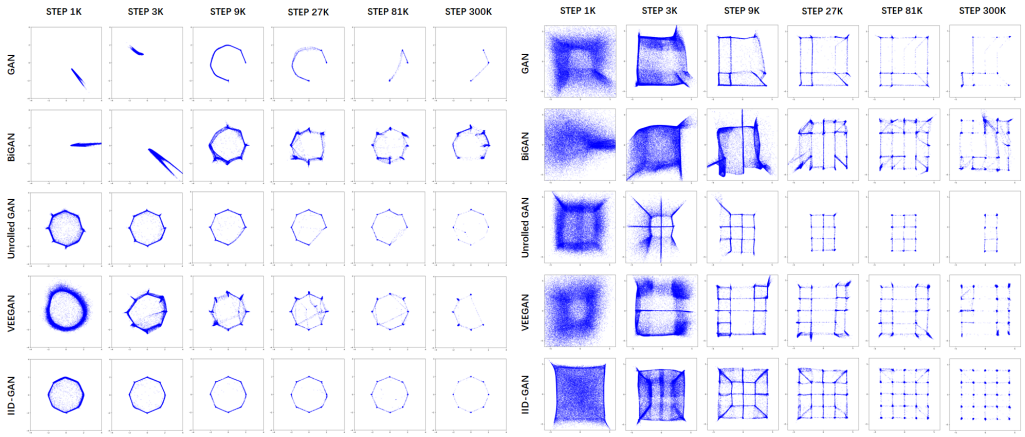
The generation results for Ring and Grid data compared with other methods are given in Figure 10(a) and Figure 10(b).

Results of M-D different Gaussion loss are presented in Figure 9.



(a) Generation quality (b) Inverse KL divergence

Figure 9: Comparison of Gaussian consistency loss on Ring data.



(a) Results of Ring data (b) Results of Grid data

Figure 10: Comparison among different methods for Ring and Grid.

G REAL DATA

G.1 NETWORK ARCHITECTURES FO REAL DATA

Table 10: Network Architecture of Inverse F for CIFAR-10 and CIFAR-100.

Layer	Output size	Activation	BN
Conv2d	$16, 16, 3$	ReLu	Yes
Conv2d	$8, 8, 64$	ReLu	Yes
Conv2d	$8, 8, 128$	ReLu	Yes
Flatten	-	-	-
Linear	100	-	-

Table 11: Network Architecture of Generator G for CIFAR-10 and CIFAR-100.

Layer	Output size	Activation	BN
Linear	16384	Relu	Yes
Conv'2d	$8, 8, 128$	ReLu	Yes
Conv'2d	$16, 16, 64$	ReLu	Yes
Conv'2d	$32, 32, 3$	Tanh	Yes

Table 12: Network Architecture of Discriminator D for CIFAR-10 and CIFAR-100.

Layer	Output size	Activation	BN
Conv2d	$16, 16, 64$	LeakyReLu	No
Conv2d	$8, 8, 128$	LeakyReLu	Yes
Conv2d	$4, 4, 256$	LeakyReLu	Yes
Flatten	-	-	-
Linear	1	-	-

Table 13: Network Architectures of Generator G for STL-10.

Layer	Output size	Kernel
Linear	8192	
ResnetBlock	4×4	3×3, 256 3×3, 256 1×1, 256
Upsample	8×8	scale factor = 2.0
ResnetBlock	8×8	3×3, 128 3×3, 128 1×1, 128
Upsample	16×16	scale factor = 2.0
ResnetBlock	16×16	3×3, 64 3×3, 64 1×1, 64
Upsample	32×32	scale factor = 2.0
ResnetBlock	32×32	3×3, 64 3×3, 64
Conv2d	32×32	

Table 14: Network Architecture of Inverse F for STL-10.

Layer	Output size	Kernel
Conv2d	32×32	$3 \times 3, 64$
ResnetBlock	32×32	3×3, 64 3×3, 64
AvgPool2d	16×16	3×3, stride 2
ResnetBlock	16×16	3×3, 64 3×3, 128 1×1, 128
AvgPool2d	8×8	3×3, stride 2
ResnetBlock	8×8	3×3, 128 3×3, 256 1×1, 256
AvgPool2d	4×4	3×3, stride 2
ResnetBlock	4×4	3×3, 256 3×3, 512 1×1, 512
Linear	20	

G.2 RESULTS FOR REAL DATA

Results on disentanglement. We perform unsupervised disentanglement learning with M-D IID-GAN as shown in Fig. 12. We study it with polar coordinates system and found that the disentanglement near the Gaussian origin is poor, and it is better if sampling is far from the origin point (i.e. the area with a larger polar radius). By varying polar radius and polar angle, we obtain a good disentanglement result. As shown in Figure 11, we can see the disentanglement results with different coordinate systems. By varying polar angle (y-axis), we can get unsupervised disentanglement results with a large polar radius (x-axis).

Condtional GAN Results. in this experiment, we chose relatively low latent dimensions in order to make the generation more challenging and to make the mode collapse contrast more obvious.

Table 15: Network Architecture of Discriminator D for STL-10.

Layer	Output size	Kernel
Conv2d	32×32	$3 \times 3, 64$
ResnetBlock	32×32	$3 \times 3,\ 64$ $3 \times 3,\ 64$
AvgPool2d	16×16	3×3, stride 2
ResnetBlock	16×16	$3 \times 3,\ 64$ $3 \times 3,\ 128$ $1 \times 1,\ 128$
AvgPool2d	8×8	3×3, stride 2
ResnetBlock	8×8	$3 \times 3,\ 128$ $3 \times 3,\ 256$ $1 \times 1,\ 256$
Linear	10	
Linear	1	

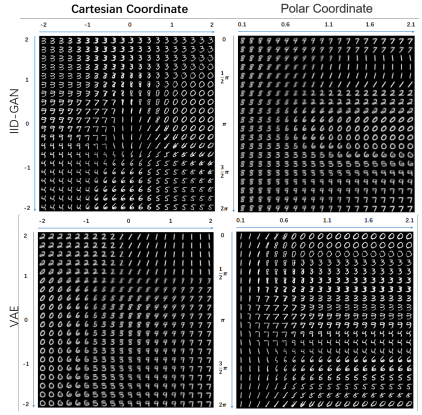


Figure 11: Uniformly sampling in different Coordinate systems for MNIST with IID-GAN and VAE model.

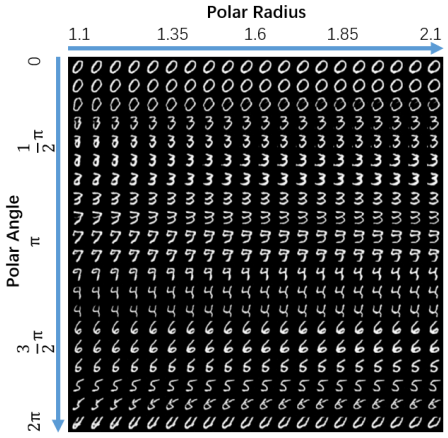


Figure 12: Unsupervised disenchantment with uniform sampling from the polar coordinate system by IID-GAN. By varying polar angle (y-axis), we can get unsupervised disentanglement results with a large polar radius (x-axis). See more details in appendix.

Here we use DCGAN network for generation. The generation results for CIFAR-10 with the latent dimension equal to 5 and 10 are shown in Figure 14 and Figure 15 and the generation results for MNIST with the latent dimension equal to 2 are shown in Figure 13.

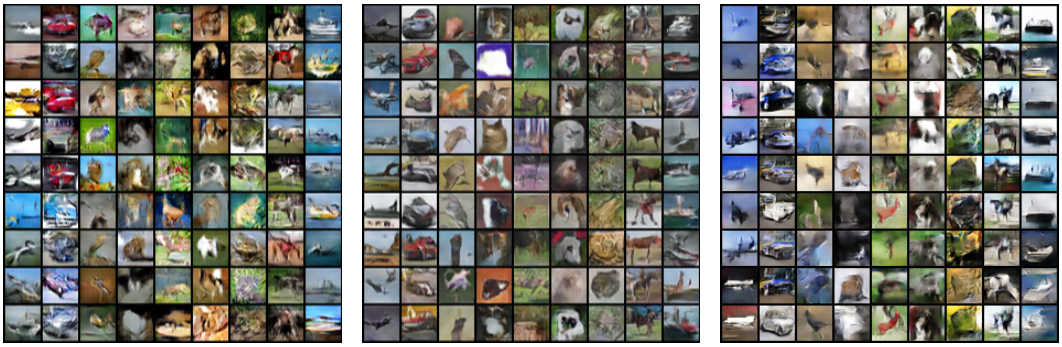
On the MNIST dataset, due to the low complexity of the images, we are able to generate recognizable images with 2-dimensional latent input z, which allows us to correspond the image to the two-dimensional z-plane. To observe the distribution and diversity characteristics of the generated images, we averaged 20 points between -2 and 2 for each dimension of z. The images were generated and arranged according to the distribution of z as shown in Figure 16. To further explore the role of our framework for generating diversity on CIFAR10 dataset, we change a dimension of the input latent code z by increasing or decreasing the value by gradient and use this z to generate, producing a gradual series of images, and we can observe that compared to the similarity of the images generated by the original CGAN during the change of the input latent code z, our model is able to capture more patterns in a single dimension, improving the generative diversity and presenting a decoupling effect to some extent. We select three image labels for each model under each number of z-dimension for display, and the image results can be found in Figure 17 and Figure 18.

(a) Conditional GAN (b) IID-GAN(1-D) (c) IID-GAN(M-D)

Figure 13: Results of conditional IID-GAN for MNIST dataset with latent dimension equal to 2. The configuration is the same as Figure 14. Although the images express same Arabic numbers, our model generates these numbers with significantly more patterns, reflecting the promotion of diversity.

(a) Conditional GAN (b) MSGAN (c) IID-GAN(M-D)

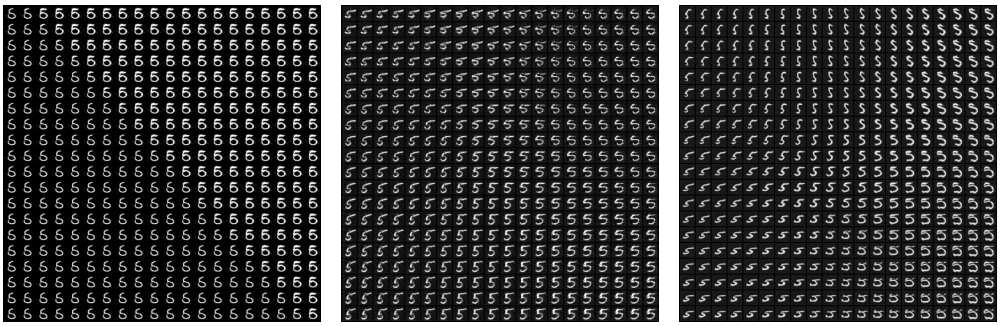
Figure 14: Results of conditional IID-GAN for CIFAR-10 dataset. Different columns represent different labels. We use the configuration with a batchsize of 256 and a learning rate of 0.0002 here.



(a) Conditional GAN (b) IID-GAN(1-D) (c) IID-GAN(M-D)

Figure 15: Results of conditional IID-GAN for CIFAR-10 dataset with random sampling on labels and z.

Unconditional Results. As shown in Figure 19, we can see that our IID-GAN can cover almost all modes in StackedMNIST datasets. Figure 20 and Figure 21 are the generation results for two IID-GAN models in CIFAR10 and CIFAR100.

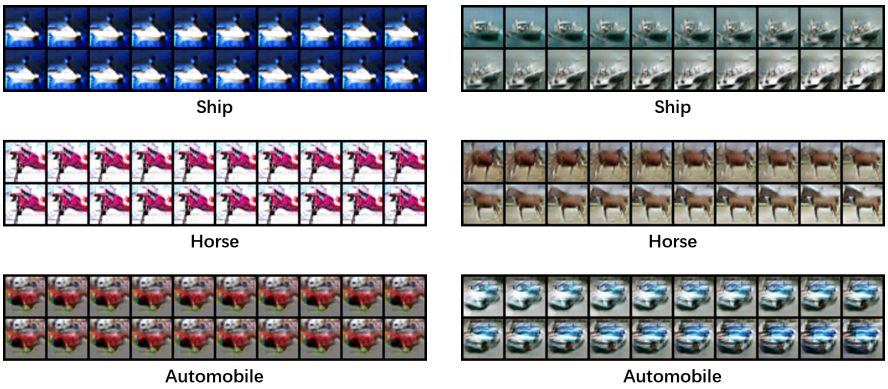


(a) Conditional GAN (b) IID-GAN(1-D) (c) IID-GAN(M-D)

Figure 16: The generated distribution of images in the two-dimensional z-plane, with both dimensions of z taking values from -2 to 2. The configuration is the same as Figure 13. It can be seen that IID-GAN have a better diversity performance compared to CGAN.

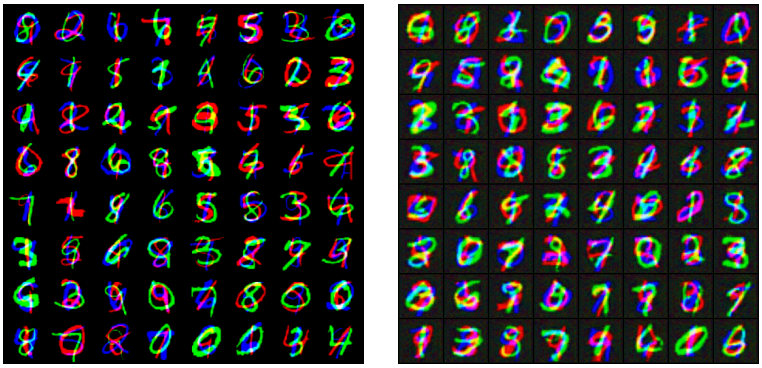
(a) Conditioned GAN (b) Conditioned IID-GAN(1-D) (c) Conditioned IID-GAN(M-D)

Figure 17: Comparison on latent dimension of 10 on CIFAR10 dataset with one dimension value of the input latent code z increasing by gradient. The configuration is the same as Figure 14.



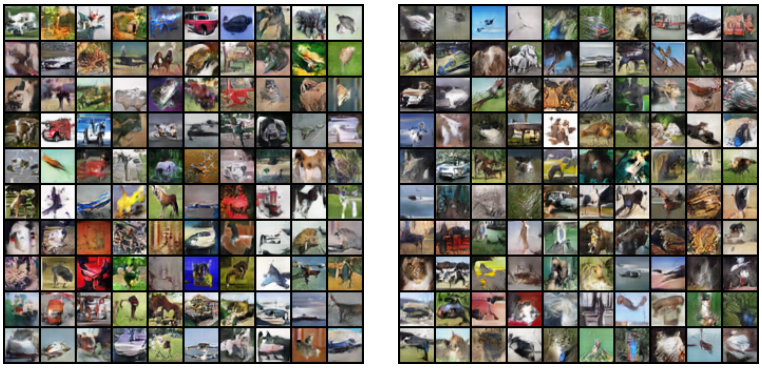
(a) Conditional GAN (b) IID-GAN(M-D)

Figure 18: Comparison on latent dimension of 5 on CIFAR10 dataset with one dimension value of the input latent code z increasing by gradient. The configuration is the same as Figure 14.



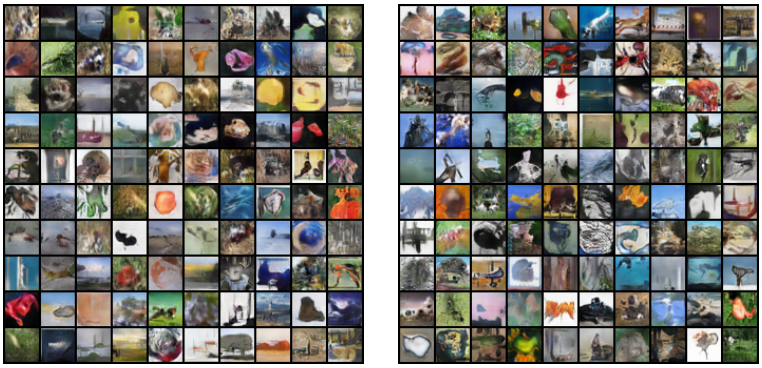
(a) Ground truth (b) IID-GAN

Figure 19: Generation results on StackedMNIST for unconditional IID-GAN.



(a) IID-GAN(1-D) (b) IID-GAN(M-D)

Figure 20: Generation results on CIFAR-10 for unconditional IID-GAN.



(a) IID-GAN(1-D) (b) IID-GAN(M-D)

Figure 21: Generation results on CIFAR-100 for unconditional IID-GAN.