Unsupervised Image Generation with Infinite Generative Adversarial Networks

Hui Ying\(^1\), He Wang\(^2\), Tianjia Shao\(^1\), Yin Yang\(^3\), Kun Zhou\(^1\)
\(^1\)Zhejiang University \(^2\)University of Leeds \(^3\)Clemson University
huiying@zju.edu.cn, H.E.Wang@leeds.ac.uk, tjshao@zju.edu.cn, yin5@clemson.edu, kunzhou@acm.org

Abstract

Image generation has been heavily investigated in computer vision, where one core research challenge is to generate images from arbitrarily complex distributions with little supervision. Generative Adversarial Networks (GANs) as an implicit approach have achieved great successes in this direction and therefore been employed widely. However, GANs are known to suffer from issues such as mode collapse, non-structured latent space, being unable to compute likelihoods, etc. In this paper, we propose a new unsupervised non-parametric method named mixture of infinite conditional GANs or MIC-GANs, to tackle several GAN issues together, aiming for image generation with parsimonious prior knowledge. Through comprehensive evaluations across different datasets, we show that MIC-GANs are effective in structuring the latent space and avoiding mode collapse, and outperform state-of-the-art methods. MIC-GANs are adaptive, versatile, and robust. They offer a promising solution to several well-known GAN issues. Code available: github.com/yinghdb/MICGANs.

1. Introduction

GANs have achieved great successes in a fast-growing number of applications [19]. The success lies in their ability to capture complex data distributions in an unsupervised, non-parametric and implicit manner [13]. Yet, such ability comes with limitations, such as mode collapse. Despite a range of methods attempting to address these issues, they are still open. This motivates our research aiming to mitigate several limitations collectively including mode collapse, unstructured latent space, and being unable to compute likelihoods, which we hope will facilitate follow-up GAN research and broaden their downstream applications.

GANs normally consist of two functions: a generator and a discriminator. In image generation, the discriminator distinguishes between real and generated images, while the generator aims to fool the discriminator by generating images that are similar to real data. The widely known mode collapse issue refers to the generator’s tendency to only generate similar data which aggregate around one or few modes in a multi-modal data distribution, e.g., only generating cat images in a cat/dog dataset. There has been active research in distribution matching to solve/mitigate mode collapse [31, 45, 50, 55], which essentially explicitly/implicitly minimizes the distributional mismatch between the generated and real data. In parallel, it is found that latent space structuring can also help, e.g. by introducing conditions [39], noises [23], latent variables [5] or latent structures [15]. In comparison, latent space structuring does enable more downstream applications such as controlled image generation, but they normally require strong prior knowledge of the data/latent space structure, such as class labels or the cluster number in the data or the mode number in the latent space. In other words, they are either supervised, or unsupervised but parametric and prescribed.

We simultaneously tackle the latent space structure and mode collapse by proposing a new, unsupervised and non-parametric method, mixture of infinite conditional GANs or MIC-GANs. Without loss of generality, we assume an image dataset contains multiple (unlabelled) clusters of images, with each cluster naturally forming one mode. Instead of making a GAN avoid mode collapse, we make use of it, i.e. exploiting GAN’s mode collapse property, to let one GAN cover one mode so that we can use multiple GANs to capture all modes. Next, doing so naturally brings the question of how many GANs are needed. Instead of relying on the prior knowledge [3, 15], we aim to learn the number of GANs needed from the data. In other words, MIC-GANs model the distribution of an infinite number of GANs. Meanwhile, we also construct a latent space according to the data space by letting each GAN learn to map one latent mode to one data mode. Since there can be an infinite number of modes in the data space, there are also the same number of modes in the latent space, each associated with one GAN. The latent space is then represented by a convex combination of GANs and is therefore structured.

To model a distribution of GANs, our first technical novelty is a new Bayesian treatment on GANs, with a family...
of non-parametric priors on GAN parameters. Specifically, we assume an infinite number of GANs in our reservoir, so that for each image, there is an optimal GAN to generate it. This is realized by imposing a Dirichlet Process [11] over the GAN parameters, which partitions the probabilistic space of GAN parameters into a countably infinite set where each element corresponds to one GAN. The image generation process is then divided into two steps: first choose the most appropriate GAN for an image and then generate the image using the chosen GAN.

Our second technical novelty is a new hybrid inference scheme. Training MIC-GANs is challenging due to the infinity nature of DP. Not only do we need to estimate how many GANs are needed, we also need to compute their parameters. Some specific challenges include: 1) unable to compute likelihoods from GANs (a fundamental flaw of GANs) [9]; 2) lack of an explicit form of GAN distributions; 3) prohibitive computation for estimating a potentially infinite number of GANs. These challenges are beyond the capacity of existing methods. We therefore propose a new hybrid inference scheme called Adversarial Chinese Restaurant Process.

MIC-GANs are unsupervised and non-parametric. They automatically learn the latent modes and map each of them to one data mode through one GAN. MIC-GANs not only avoid mode collapse, but also enable controlled image generation, interpolation among latent modes, and a systematic exploration of the entire latent space. Through extensive evaluation and comparisons, we show the superior performance of MIC-GANs in data clustering and generation.

2. Related Work

Mode Collapse in GANs GANs often suffer from mode collapse, where the generator learns to generate samples from only a few modes of the true distribution while missing many other modes. To alleviate this problem, researchers have proposed a variety of methods such as incorporating the minibatch statistics into the discriminator [53], adding regularization [4, 59], unrolling the optimization of the discriminator [38], combining a Variational Autoencoder (VAE) with GANs using variational inference [50], using multiple discriminators [7], employing the Gaussian mixture as a likelihood function over discriminator embeddings [10], and applying improved divergence metrics in the loss of discriminator [2, 14, 37, 40, 45]. Other methods focus on minimizing the distributional mismatch between the generated and real data. For example, VEEGAN [55] introduces an additional reconstructor network to enforce the bijection mapping between the true data distribution and Gaussian random noise. MMD GAN [31] is proposed to align the moments of two distributions with generative neural networks. Most of existing methods essentially map one distribution (often Gaussian or uniform) to a data distribution with an arbitrary number of modes. This is an extremely challenging mapping to learn, leading to many issues such as convergence and inability to learn complex distributions [46]. Rather than avoiding mode collapse, we exploit it by letting one GAN learn one mode in the data distribution (assuming one GAN can learn one mode), so that we can use multiple GANs to capture all modes in the data distribution. This not only naturally avoids mode collapse but leads to more structured latent space representations.

Latent Space Structure in GANs Early GANs focus on mapping a whole distribution (e.g., uniform or Gaussian) to a data distribution. Since then, many efforts have been made to structure the latent space in GANs, so that the generation is controllable and the semantics can be learned. Common strategies involve introducing conditions [36, 39, 41, 47, 58], latent variables [5], multiple generators [12, 32], noises [23, 24] and clustering [42]. Recent approaches also employ mixture of models (e.g., Gaussian mixture models) to explicitly parameterize the latent space [3, 15]. However, these methods usually require strong prior knowledge, e.g., class labels, the cluster number in the data distribution and the mode number in the latent space, with prescribed models to achieve the best performance. In this paper, we relax the requirement of any prior knowledge of the latent/data space. Specifically, MIC-GANs are designed to learn the latent modes and the data modes simultaneously and automatically. This is realized by actively constructing latent modes while establishing a one-to-one mapping between latent modes and data modes, where each GAN learns one mapping. Consequently, the latent space is structured by a convex combination of GANs. DMGAN [25] is the most similar work to ours, which employs multiple generators to learn distributions with disconnected support without prior knowledge. In contrast, MIC-GANs neither impose any assumption on the connectivity of the support, nor require multiple generators. Besides, MIC-GANs have a strong clustering capability for learning the latent modes. An alternative approach is to use Variational Autoencoder (VAE) which can structure the latent space (i.e., a single Gaussian or mixed Gaussians) during learning [8, 20, 21, 26, 43, 56], but they often fail to generate images with sharp details. We therefore focus on GANs.

3. Methodology

3.1. Preliminary

Given image data $X$, a GAN can be seen as two distributions $G(X | \theta^g, Z)$ and $D([0, 1]|\theta^d, X)$, with $\theta = [\theta^g, \theta^d]$ being the network weights and $Z$ being drawn from a distribution, e.g., Gaussian. $\theta$ uniquely defines a GAN. Unlike traditional GANs, we use a Bayesian approach and treat $\theta$ as random variables which conform to some prior distribution parameterized by $\Phi = [\Phi^g, \Phi^d]$. The inference of $\theta$ can
be conducted by iteratively sampling [51]:

\[ p(\theta^d|Z, \theta^d) \propto (\prod_{i=1}^{N^o} D(G(z_{i|}\theta^d); \theta^d)) p(\theta^d|\Phi) \]  
(1)

\[ p(\theta^d|Z, X, \theta^d) \propto \prod_{i=1}^{N^d} D(x_{i|} ; \theta^d) \times \prod_{i=1}^{N^o} \left(1 - D(G(z_{i|}\theta^d); \theta^d)\right) \times p(\theta^d|\Phi^d) \]  
(2)

where \( N^d \) and \( N^o \) are the total numbers of generated and real images, \( p(\theta^d|\Phi^d) \) and \( p(\theta^d|\Phi^d) \) are prior distributions of the network weights and sometimes combined as \( p(\theta|\Phi) \) for simplicity. For our goals, the choice of the prior is based on the following consideration. First, if \( X \) has \( K \) modes corresponding to \( K \) clusters, we aim to learn \( K \) mappings through \( K \) distinctive GANs, and each GAN is only responsible for generating one cluster of images. This dictates that the draws from the prior needs to be discrete. Second, since the \( K \) value is unknown \( a \) priori, we need to assume that \( K \to \infty \). Therefore, we employ a Dirichlet Process (DP) as the prior \( p(\theta|\Phi) \) for \( \theta \).

A DP \((\alpha, \Phi)\) is a distribution of probabilistic distributions, where \( \alpha \) is called concentration and \( \Phi \) is the base distribution. It describes a 'rich get richer' sampling [44]:

\[ \theta_i|\theta_{i-1}, \alpha, \Phi \sim \sum_{i=1}^{\infty} \frac{1}{i - 1 + \alpha} \delta_{\theta_i} + \frac{\alpha}{i - 1 + \alpha} \Phi \]  
(3)

where an infinite sequence of random variables \( \theta_i \) are i.i.d. according to \( \Phi \). \( \theta_{i-1} = \{\theta_1, \ldots, \theta_{i-1}\} \). \( \delta_{\theta_i} \) is a delta function at a previously drawn sample \( \theta_i \). When a new \( \theta_i \) is drawn, either a previously drawn sample is drawn again (with a probability proportional to \( \frac{1}{i - 1 + \alpha} \)), or a new sample is drawn (with a probability proportional to \( \frac{\alpha}{i - 1 + \alpha} \)). Assuming each \( \theta \) has a value \( \phi \) there can be multiple \( \theta \)s having the same value \( \phi \). So there are only \( K \) distinctive values in a total of \( i \) samples drawn so far in Equation 3 where \( K < i \). An intuitive (but not rigorous) analogy is rolling a dice multiple times. Each time one side (a sample) is chosen but overall there are only \( K = 6 \) possible values.

To see the 'rich get richer' property, the more \( \phi \) has been drawn before, the more likely it will be drawn again. This property is highlighted by another equivalent representation called Chinese Restaurant Processes (CRP) [1], where the number of times the \( k \)th \((k \in K)\) value \( \phi_k \) has been drawn is associated with its probability of being drawn again:

\[ \theta_i|\theta_{i-1}, \alpha, \Phi \sim \sum_{k=1}^{K} \frac{N_k}{i - 1 + \alpha} \delta_{\phi_k} + \frac{\alpha}{i - 1 + \alpha} \Phi \]  
(4)

where \( \delta_{\phi_k} \) is a delta function at \( \phi_k \), and \( N_k \) is how many times that \( \phi_k \) has been sampled so far. Equation 3 and 4 are equivalent with the former represented by draws and the latter by actual values.

### 3.2. Mixture of Infinite GANs

We propose a new Bayesian GAN which is a mixture of infinite GANs model. Following Eq. 4, \( \phi \) represents the network weights of a GAN. Imagine we have \( K \to \infty \) GANs and examine \( X \) one by one. For each image \( x_i \), we sample the best GAN \( \phi_{c_i} \) (based on some criteria) to generate it. So \( N_k = \sum c_i = k \) is the total number of images already selecting the \( k \)th GAN. The more frequently a GAN is selected, the more likely it will be selected in future. If all \( N_k \)s are small, then a new GAN is likely to be sampled based on \( \Phi \).

We describe the generative process of our model as:

- sample \( z_i \sim Z, \{\phi_1, \ldots, \phi_k, \ldots, \phi_K\} \sim \Phi \)
- sample \( c_i \sim CRP(\alpha, \Phi; c_1, \ldots, c_{i-1}) \), where \( c_i = k \)
- \( x_i = G(z_i; \phi_k^c) \) so that \( D(x_i; \phi_k^c) = 1 \)

where \( c_i \) now is an indicator variable, \( \phi(c_i=k)=[\phi_k^c, \phi_k^d] \) are the parameters of the \( k \)th GAN. Combining Equation 4-5 with 1-2, the inference of our new model becomes:

\[ p(\phi|\Phi) = p(c_i|e_{i-1}) \sim CRP(\alpha, \Phi, e_{i-1}) \ c \in [1, K] \]  
(6)

\[ p(\phi_k^c|Z, \phi_k^d) \propto \prod_{i=1}^{N_k^d} D(G(z_{i|}; \phi_k^d), \phi_k^c)) p(\phi_k^c|\Phi^d) \]  
(7)

\[ p(\phi_k^d|Z, X, \phi_k^c) \propto \prod_{i=1}^{N_k^d} D(x_{i|}; \phi_k^c) \times \prod_{i=1}^{N_k^d} (1 - \prod_{i=1}^{N_k^d} D(G(z_{i|}; \phi_k^d), \phi_k^c)) \]  
(8)

where \( e_{i-1} = \{c_1, \ldots, c_{i-1}\} \). Sampling \( c \) in Equation 6 will naturally compute the right value for \( K \), essentially conducting unsupervised clustering [44].

**Classical GANs as maximum likelihood.** Equation 6-8 is a Bayesian generalization of classic GANs. If a uniform prior is used for \( \Phi \) and iterative maximum a posteriori (MAP) optimization is employed instead of sampling the posterior, then the local minima give the standard GANs [13]. However, even with a flat prior, there is a big difference between Bayesian marginalization over the whole posterior and approximating it with a point mass in MAP [51]. Equation 6-8 is a specification of Equation 1-2 with a CRP prior. Although Bayesian generalization over GANs have been explored before, we believe that this is first time a family of non-parametric Bayesian priors have been employed in modeling the distribution of GANs. Further, MIC-GANs aim to capture one cluster of images with one GAN in an unsupervised manner. The CRP prior can automatically infer the right value for \( K \) instead of pre-defining one as in existing approaches [33, 42, 15], where overestimating \( K \) will divide a cluster arbitrarily into several ones while underestimating \( K \) will mix multiple clusters.
3.3. Mixture of Infinite Conditional GANs

Iterative sampling over Equation 6-8 would theoretically infer the right values for \( \phi^a \) and \( K \). While \( \phi^d \) and \( \phi^q \) can be sampled [51] or approximated [13], \( K \) needs to be sampled indirectly by sampling \( c \), which turns out to be extremely challenging. To see the challenges, we need to analyze the full conditional distribution of \( c \). To derive the full distribution, we first give the conditional probability of \( c \) based on Eq. 4 as in [44]:

\[
p(c_i = k | c_{-i}) \propto \frac{N_k}{i - 1 + \alpha} \quad \text{for all } j < i
\]

Given the likelihood \( p(x_i | z_i, \phi_k) \), or \( p(x_i | \phi_k) \) if we omit \( z_i \) as it is independently sampled, we now combine it with Equation 9 to obtain the full conditional distribution of \( c_i \) given all other indicators and the current \( x_i \):

\[
p(c_i = k | c_{-i}, x_i, \phi) \propto \frac{N_k}{i - 1 + \alpha} p(x_i | \phi_k) \quad \text{if } \phi_k \text{ exists},
\]

\[
p(c_i = c_{new} | c_{-i}, x_i, \phi) \propto \frac{\alpha}{i - 1 + \alpha} \int p(x_i | \phi) d\phi
\]

if a new \( \phi_{new} \) is needed

where if a \( \phi_{new} \) is needed then it will be sampled from the posterior \( p(\phi | x_i, \Phi) \). Eq. 10 is used to sample Eq. 6.

3.3.1 Challenges of Inference

One method to sample \( c_i \) is Gibbs sampling [44], which requires the prior \( p(\phi | \Phi) \) to be a conjugate prior for the likelihood; otherwise additional sampling (e.g. Metropolis-Hasting) is needed to approximate the integral in Equation 10. However, for MIC-GANs, not only is the prior not a conjugate prior for the likelihood, neither the likelihood nor the posterior can be even explicitly represented, which bring the following challenges: (1) Unable to directly compute the likelihood \( p(x_i | \phi_c) \), which is a well-known issue for GANs [9]. (2) The sampling from \( p(\phi | x_i, \Phi) \) is ill-conditioned. Since the prior \( \Phi \) cannot be explicitly represented, direct sampling from \( p(\phi | x_i, \Phi) \) becomes impossible. Alternatively, methods such as Markov Chain Monte Carlo are theoretically possible. But the dimension is often high, which will make the sampling prohibitively slow. (3) Sampling will dynamically change \( K \). Each time \( K \) grows/shrinks, a GAN needs to be created/destroyed, which is far from ideal in terms of the training speed. This is also an issue of existing methods with multiple GANs [30, 17].

3.3.2 Enhanced Model for Inference

To tackle challenge (2), we introduce a conditional variable \( C \) while forcing all GANs to share the same \( \phi \) so that they become \( G_{\phi^s}(X|Z, C_k) \) and \( D_{\phi^s}([0, 1]|X, C_k) \) instead of \( G(X|Z; \phi^s_k) \) and \( D([0, 1]|X, \phi^q) \) respectively, where \( C \sim p(C) \) is well-behaved, e.g. a multivariate Gaussian. This formulation is similar to Conditional GANs (CGANs) but with a Bayesian treatment on \( C \). Indeed, by introducing a conditional variable into multiple layers in the network, we exploit its ability of changing the mapping. Also, we now only need one GAN parameterized by \( \phi = [\phi^q, \phi^d] \) and eliminate the need for multiple GANs without compromising the ability of learning \( K \) distinctive mappings. Now the role of \( C \) is the same as \( \Phi \) in Equation 5, leading to:

\[
\text{sample } z_i \sim Z, \{C_0, \ldots, C_k, \ldots, C_K\} \sim p(C)
\]

\[
\text{sample } c_i \sim CRP(\alpha, C; c_1, \ldots, c_{i-1}), \text{ where } c_i = k
\]

\[
x_i = G_{\phi^s}(z_i, C_k) \text{ so that } D_{\phi^s}(x_i, C_k) = 1,
\]

where sampling from the posterior \( p(\phi | x_i, C) \) (previously \( p(\phi | x_i, \Phi) \) in Equation 10 becomes feasible. Note our formulation is different from traditional CGANs and GANs with multiple generators [17, 52, 28, 33] in: (1) our approach is still Bayesian. (2) we still model an infinite number of GANs and do not rely or impose assumptions on the prior knowledge of cluster numbers. (3) we do not need to actually instantiate multiple GANs.

Next, we still need to be able to compute the likelihood \( p(x_i | \phi_k) \) in Equation 10, which is challenge (1). Now \( p(x_i | \phi_k) \) becomes \( p(x_i | C_k) \). Since GANs cannot directly compute likelihoods, we employ a surrogate model that can compute likelihoods while mimicking GAN’s mapping behavior. Each \( C \) corresponds to one cluster, so the GANs can be seen as a mapping between Cs and images. This correspondence is exactly the same as classifiers. So we design a classifier \( Q \) to learn the mapping so that \( p(x_i | C_k) \propto Q(c = k | x_i), \) where \( c \) is the same \( c \) as in Equation 10 but here is also a cluster label out of \( K \) clusters. Existing image classifiers can approximate likelihoods, e.g. through softmax. However, our experiments show that softmax-based likelihood approximation tends to behave like discriminative classifiers which focus on learning the classification boundaries. In Equation 10, we need a classifier with density estimation. We thus define \( Q \) as:

\[
Q(c_i = k | x_i, \phi^q) \propto p(x_i | C_k, \phi^q)p(C_k | C)p(\phi^q | \Phi)
\]

\[
= N(x_i | \mu_k, \Sigma_k, \phi^q)p(C_k | C)p(\phi^q | \Phi)
\]

\[
= N'(y_i | \mu_k, \Sigma_k)p(y_i | x_i, \phi^q)p(C_k | C)p(\phi^q | \Phi)
\]

where \( N \) is a Normal distribution. \( \mu_k \) and \( \Sigma_k \) are the mean and covariance matrix of the \( k \)th Normal. \( Q \) is realized by a deep neural net, parameterized by \( \phi^q \) with a prior \( p(\phi^q | \Phi) \), and classifies the latent code \( y_i \) of \( x_i \) by a Infinite Gaussian Mixture Model (IGMM) [49]. To see this is an IGMM, \( p(C_k | C) \) is the same CRP as in Equation 11, so that now the
suitable for finding the posterior distributions of \( \theta \) and MAP optimization on \( s \) and Gibbs sampling. While the former is suitable for two schemes: optimization based on stochastic gradient descent is mainly learned through the GAN training. In other words, the shape of \( p(C) \) is not essential, as long as \( C \)s are distinctive, i.e. conditioning different mappings. So we fix \( p(C) \) to be a multivariate Gaussian without compromising the modeling power of MIC-GANs.

Finally, since the learned GANs are governed by a DP, another equivalent representation is \( G(X|Z,C) = \sum_{k=1}^{\infty} \beta_k G_k(X|Z,C_k) \), where the subscript \( k \) indicates the \( k \)th GAN. \( \beta_k \) is a weight and \( \sum_{k=1}^{\infty} \beta_k = 1 \). This is another interpretation of DP called stick-breaking where \( \beta_k = v_k \prod_{j=1}^{k-1} v_j \) and \( v \sim \text{Beta} \). After learning, the weights \( \beta_k \)s can be computed by the probability of images assigned to each GAN. The stick-breaking interpretation indicates that the learned Gs form a convex combination of clusters, which is an active construction of the latent space. This enables controlled generation e.g. using single \( C \)s to generate images in different clusters, and easy exploration of the data space e.g. via interpolating \( C \)s.

\[ p(\phi^g|X, \mu, \Sigma) \approx \prod_{i=1}^{N} \left( \sum_{k=1}^{K} N(y_i|\mu_k, \Sigma_k)p(y_i|x_i, \phi^g) \right) \times p(\phi^g|\Phi^g) \]

\[ p(\mu, \Sigma, x_i, c_i = k, x_{-i}, c_{-i}, \phi^g) = \int \int p(y_i|\mu_k, \Sigma_k) \]

\[ p(y_i|x_i, \phi^g)p(\mu_k, \Sigma_k|x_{-i}, c_{-i})d\mu_k d\Sigma_k \]

where \( N \) is total number of images, \( K \) is total number of clusters, \( x_{-i} \) is all images except \( x_i \) with \( c_{-i} \) as their corresponding indicators. A Normal-Inverse-Wishart prior can be imposed on \( \mu, \Sigma \) and we refer the readers to [49] for details. Alternatively, we could use IGMM directly on the images, but better classification results are achieved by classifying their latent features, as in the standard setting of many deep neural network based classifiers. We realize \( p(y_i|x_i, \phi^g) \) as an encoder in \( Q \).

Lastly for challenge (3), to avoid dynamically growing and shrinking \( K \), we employ a truncated formulation inspired by [18], where an appropriate truncation level can be estimated. Essentially, the formulation requires us to start with a sufficiently large \( K \) then learn how many modes are actually needed, which is automatically computed due to the aggregation property of DPs. Note that the truncation is only for the inference purpose and does not affect the capacity of MIC-GAN to model an infinite number of GANs. We refer the readers to [18] for the mathematical proofs.

3.4. Adversarial Chinese Restaurant Process

Finally, we have our full MIC-GANs model (Eq. 11-12) and are ready to derive our new sampling strategy called Adversarial Chinese Restaurant Process (ACRP). A Bayesian inference can be done through Gibbs sampling: iteratively sampling on \( \phi^g, \phi^d, c \) (Equation 6-8, 10), \( \mu, \Sigma \) and \( \phi^d \) (Equation 12-14). However, this tends to be slow given the high dimensionality. We thus propose to combine two schemes: optimization based on stochastic gradient descent and Gibbs sampling. While the former is suitable for finding the local minima that are equivalent to using flat priors and MAP optimization on \( [\phi^g, \phi^d, \phi^d] \) [51], the latter is suitable for finding the posterior distributions of \( c, \mu \) and \( \Sigma \). We give the general sampling process of ACRP in Alg.1 and refer the readers to the supplementary materials for details.

In non-parametric Bayesian, \( p(C) \) is governed by hyper-parameters which can be incorporated into the sampling. However, unlike traditional non-parametric Bayesian where a \( C \) would specify a generation process, our generation process is mainly learned through the GAN training. In other

\begin{algorithm}
\caption{Adversarial Chinese Restaurant Process}
\begin{algorithmic}
\Require
\Statex \textit{epochs} - the number of total training epochs;
\Statex N is the total number of images;
\Statex \text{Initialize all variables (supplementary material)};
\For {\text{epoch } = 1 \text{ to } \text{epochs}}
\State For \( x_i \) in \( X \), classify \( x_i \) via Eq. 12;
\State Compute \( \{N_k\}_{k=1}^{K} \) via Eq. 10;
\State For \( k = 1 \) to \( K \) Sample \( \mu_k \) and \( \Sigma_k \) via Eq. 14;
\State Sample \( \{c_i\}_{i=1}^{N} \) via Eq. 10;
\State Optimize \( \phi^g \) and \( \phi^d \) via conditional GAN loss;
\State Optimize \( \phi^d \) via Maximum Likelihood (Eq. 13);
\EndFor
\end{algorithmic}
\end{algorithm}

Implementation details. Due to the space limit, please refer to the supplementary materials for the details of models, data processing, training settings and performances.

4. Experiments

We adopt StyleGAN2 [24], StyleGAN2-Ada [22] and DCGAN [48] to validate that MIC-GANs can incorporate different GANs. We employ several datasets for extensive evaluation, including MNIST [29], FashionMNIST [57], CatDog from [6] and CIFAR-10 [27]. Moreover, we build a challenging dataset, named Hybrid, consisting of data with distinctive distributions. It is a mixture of the ’0’, ’2’, ’4’, ’6’, ’8’ from MNIST, the ’T-shirt’, ’pullover’, ’trouser’, ’bag’, ’sneaker’ from FashionMNIST, the cat images from Catdog and human face images from CelebA [34].

For comparison, we employ as baselines several state-of-the-art GANs including DMGAN [25], InfoGAN [5], ClusterGAN [42], DeliGAN [15], Self-Conditioned GAN [33] and StyleGAN2-Ada [22], whose code is shared by the authors. Notably, DeliGAN, InfoGAN, ClusterGAN and Self-
Table 1. Purity and FID scores from MIC-GANs.

<table>
<thead>
<tr>
<th></th>
<th>MNIST</th>
<th>FashionMNIST</th>
<th>CatDog</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purity</td>
<td>0.9489</td>
<td>0.6362</td>
<td>0.9736</td>
<td>0.9567</td>
</tr>
<tr>
<td>FID</td>
<td>12.79</td>
<td>17.97</td>
<td>26.24</td>
<td>11.2</td>
</tr>
</tbody>
</table>

Table 2. Purity and FID of MIC-GANs(1), DMGAN(2), InfoGAN(3), ClusterGAN(4) and DeliGAN(5).

<table>
<thead>
<tr>
<th>K</th>
<th>Purity</th>
<th>FID</th>
<th>K</th>
<th>Purity</th>
<th>FID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.9489</td>
<td>12.79</td>
<td>25</td>
<td>0.9567</td>
<td>11.2</td>
</tr>
<tr>
<td>2</td>
<td>0.7612</td>
<td>10.82</td>
<td>25</td>
<td>0.8959</td>
<td>32.31</td>
</tr>
<tr>
<td>3</td>
<td>0.8589</td>
<td>11.9</td>
<td>12</td>
<td>0.881</td>
<td>71.84</td>
</tr>
<tr>
<td>4</td>
<td>0.8784</td>
<td>9.27</td>
<td>12</td>
<td>0.9457</td>
<td>19.15</td>
</tr>
<tr>
<td>5</td>
<td>n/a</td>
<td>127.58</td>
<td>10</td>
<td>n/a</td>
<td>241.69</td>
</tr>
</tbody>
</table>

4.1. Unsupervised Clustering

Although MIC-GANs focus on unsupervised image generation, it clusters data during learning. We evaluate its clustering ability by Cluster Purity [35] which is a common metric to measure the extent to which clusters contain a single class: \( Purity = \frac{1}{N} \sum_{i=1}^{K} \max_j |c_i \cap t_j| \), where \( N \) is the number of images, \( K \) is the number of clusters, \( c_i \) is a cluster and \( t_j \) is the class which has the highest number of images clustered into \( c_i \). The purity reaches the maximum value 1 when every image is clustered into its own class. Since MIC-GANs compute \( K \) clusters, we rank them by their \( \beta_k \)s in the descending order and choose the top \( n \) clusters for purity calculation, where \( n \) is the ground-truth class number. Note the ground-truth is only used in the testing phase, not in the training phase.

We use \( K = 20, 20, 4 \) and 25 for MNIST, FashionMNIST, CatDog and Hybrid. The results are shown in Table 1. MIC-GANs achieve high purity on MNIST, CatDog and Hybrid. On FashionMNIST, the purity is relatively low. We find that it is mainly due to the ambiguity in the class labels. One example is that the bags with/without handles are given the same class label in FashionMNIST but divided by MIC-GANs into two clusters, as shown later. This finer-level clustering is automatically conducted, which we argue is still reasonable albeit leading to a lower purity score.

As a comparison, we also evaluate the baseline methods on MNIST and Hybrid, with the ground-truth class number \( K = 10 \) for InfoGAN and ClusterGAN, and \( K = 20 \) for DMGAN. Note that DeliGAN does not have a clustering module, so we exclude it from this evaluation. For both MIC-GANs and DMGAN, we use the top \( n \) clusters where \( n \) is the ground-truth class number. As we can see in Table 2, surprisingly, MIC-GANs achieve the highest purity score (0.9489) among these methods, even without using the ground-truth class number in training. It demonstrates that the MIC-GANs can effectively learn the modes in the MNIST without any supervision. In addition, we also conduct the comparison on Hybrid, with the ground-truth \( K = 12 \) for InfoGAN and ClusterGAN, and \( K = 25 \) for DMGAN. As shown in Table 2, MIC-GANs achieve the best purity score (0.9567). ClusterGAN (0.9457) comes a close second. However, we provide the ground-truth class number to ClusterGAN which is strong prior knowledge, while MIC-GANs have no such information.

4.2. Image Generation Quality

We conduct both qualitative and quantitative evaluations on the generation quality. For quantitative evaluations, we use Frechet Inception Distance (FID) [16] as the metric. Qualitatively, generated samples can be found in Figure 1-2. Due to the space limit, we rank the modes based on their \( \beta_k \)s in the descending order and only show the top modes. More results can be found in the supplementary materials. Intuitively, all the modes are captured cleanly, shown by that the top modes contain all the classes in the datasets. This is where MIC-GANs capture most of the ‘mass’ in the datasets. In addition, each mode fully captures the variation within the mode, no matter it is the writing style in MNIST, the shape in FashionMNIST or the facial features in CatDog and CelebA. The balance between clustering and within-cluster variation is automatically achieved by MIC-GANs in an unsupervised setting. This is very challenging because the within-cluster variations are distinctive in Hybrid since the data comes from four datasets. Beyond the top modes, all the low-rank modes also capture information. But the information is less meaningful (the later modes in MNIST) or mixed (later modes in Hybrid) or contain subcategories such as the separation of bags with/without handles in FashionMNIST. However, this does not undermine MIC-GANs because the absolute majority of the data is captured in the top modes. Quantitatively, we found that good FID scores can be achieved, shown in Table 1.

We further conduct comparisons on MNIST and Hybrid using the same settings as above for all methods. As demonstrated in Table 2, MIC-GANs obtain a comparable FID score to other methods without any supervision in MNIST. Besides DeliGAN, DMGAN and MIC-GANs achieve slightly worse FID scores. We speculate that this is because MIC-GANs and DMGANs do not use prior knowledge and therefore have a disadvantage. One exception is DeliGAN. In their paper, the authors chose a small dataset (500 images) for training and achieved good results. However, when we run their code on the full MNIST dataset, we were not able to reproduce comparable results even after trying our best to tune various parameters. Next, in the chal-
Figure 1. Our results on MNIST (left) and Hybrid (right) dataset, both with \(K = 25\). Each column is generated from a mode, and the columns are sorted by \(\alpha\)s (the last 4 modes are not shown). The red boxes mark the top \(n\) modes in the results.

Figure 2. Our results on CatDog (left) and FashionMNIST (right) dataset. Each column is generated from one mode, and the columns are sorted by \(\alpha\)s.

Table 3. Comparisons of our method, SC-GAN (Self-Conditioned GAN), and StyleGAN2-Ada on CIFAR. ‘C’ means the ground-truth class number in the dataset.

Table 4. Purity and FID of ablation studies on MNIST and Hybrid.

4.3. Ablation Study

We conduct ablation studies to test MIC-GANs’ sensitivity to the free parameters. As an unsupervised and non-parametric method, there are not many tunable parameters which is another advantage of MIC-GANs. The main parameters are the \(K\) value and the GAN architecture. We therefore test another popular GAN, DCGAN [48] and vary the \(K\) value. As shown in Table 4, the purity scores are very similar, which means the clustering is not significantly affected by the choices of GANs or the \(K\) value. FID scores vary across datasets, which is mainly related to the specific choice of the GAN architecture. However, stable performance is obtained across different \(K\)s in every setting. In addition, the FID scores when \(K = 1\) are in general worse than those when \(K > 1\), confirming that our method can mitigate mode collapses. The same mode collapse mitigation can also be observed when comparing our method with StyleGAN2-Ada on CIFAR-10 (Table 3), where StyleGAN2-Ada is just our method with \(K = 1\).

4.4. Benefits of Non-parametric Learning

In real-world scenarios, we do not often know the cluster number \(a priori\), under which we investigate the performance of InfoGAN, ClusterGAN and DeliGAN. We use Hybrid and run experiments with \(K=8, 12, 16\) and \(22\) to cover \(K\) values that are smaller, equal to and bigger than the ground-truth \(K=12\). We only show the results of \(K=8\) in Figure 4 and refer the readers to the supplementary materials for fuller results and analysis. Intuitively, when \(K\) is smaller than the ground-truth, the baseline methods either cannot capture all the modes or capture mixed modes; when

---

**Table 3**: Comparisons of MIC-GANs with SC-GAN and StyleGAN2-Ada on MNIST and Hybrid datasets. The results show that MIC-GANs achieve better performance in terms of purity and FID scores compared to the baseline methods.

**Table 4**: Purity and FID scores for ablation studies on MNIST and Hybrid datasets. The scores indicate that MIC-GANs can achieve better performance across different compositions of the datasets, maintaining stable performance across different \(K\) values.

**Figure 1**: Our results on MNIST (left) and Hybrid (right) dataset, both with \(K = 25\). Each column is generated from a mode, and the columns are sorted by \(\alpha\)s (the last 4 modes are not shown). The red boxes mark the top \(n\) modes in the results.

**Figure 2**: Our results on CatDog (left) and FashionMNIST (right) dataset. Each column is generated from one mode, and the columns are sorted by \(\alpha\)s.

**Table 3**: Comparisons of our method, SC-GAN (Self-Conditioned GAN), and StyleGAN2-Ada on CIFAR. ‘C’ means the ground-truth class number in the dataset.

**Table 4**: Purity and FID of ablation studies on MNIST and Hybrid. The scores indicate that MIC-GANs can achieve better performance across different \(K\) values and compositions of the datasets.

---

**Figure 2**: Our results on CatDog (left) and FashionMNIST (right) dataset. Each column is generated from one mode, and the columns are sorted by \(\alpha\)s.
$K$ is larger than the ground-truth, they capture either mixed modes or repetitive modes. In contrast, although MIC-GANs (Figure 1-2) also learn extra modes, it concentrates the mass into the top modes resulting in clean and complete capture of modes. MIC-GANs are capable of high-quality image generation and accurate data mode capturing, while being robust to the initial guess of $K$.

### 4.5. Latent Structure

Since MIC-GANs are a convex combination of GANs, we can do controlled generation, including using a specific mode, and interpolating between two or more different modes for image generation. Figure 1-2 already show image generation based on single modes. We show interpolations between two $C$s and among four $C$s respectively in Figure 5. Through both bi-mode and multi-mode interpolation, we show that MIC-GANs structure the latent space well so that smooth interpolations can be conducted within the subspace bounded by the base modes.

### 5. Conclusion

We proposed a new unsupervised and non-parametric generative framework MIC-GANs, to jointly tackle two fundamental GAN issues, mode collapse and unstructured latent space, based on parsimonious assumptions. Extensive evaluations and comparisons show that MIC-GANs outperform state-of-the-art methods on multiple datasets. MIC-GANs do not require strong prior knowledge, nor do they need much human intervention, providing a robust and adaptive solution for multi-modal image generation.

### Acknowledgments

We thank anonymous reviewers for their valuable comments. The work was supported by NSF China (No. 61772462, No. 61890954, No. U1736217), the 100 Talents Program of Zhejiang University, and NSF under grants No. 2011471 and 2016414.


