Neural Characteristic Function Learning for Conditional Image Generation

Shengxi Li[1], Jialu Zhang[1], Yifei Li[1], Mai Xu*[1], Xin Deng[2], Li Li[1]

[1]School of Electronic and Information Engineering, Beihang University, Beijing, China
[2]School of Cyber Science and Technology, Beihang University, Beijing, China
{LiShengxi, JialuZhang, leafy, MaiXu, cindydeng, lili2005}@buaa.edu.cn

Abstract

The emergence of conditional generative adversarial networks (cGANs) has revolutionised the way we approach and control the generation, by means of adversarially learning joint distributions of data and auxiliary information. Despite the success, cGANs have been consistently put under scrutiny due to their ill-posed discrepancy measure between distributions, leading to mode collapse and instability problems in training. To address this issue, we propose a novel conditional characteristic function generative adversarial network (CCF-GAN) to reduce the discrepancy by the characteristic functions (CFs), which is able to learn accurate distance measure of joint distributions under theoretical soundness. More specifically, the difference between CFs is first proved to be complete and optimisation-friendly, for measuring the discrepancy of two joint distributions. To relieve the problem of curse of dimensionality in calculating CF difference, we propose to employ the neural network, namely neural CF (NCF), to efficiently minimise an upper bound of the difference. Based on the NCF, we establish the CCF-GAN framework to explicitly decompose CFs of joint distributions, which allows for learning the data distribution and auxiliary information with classified importance. The experimental results on synthetic and real-world datasets verify the superior performances of our CCF-GAN, on both the generation quality and stability.

1. Introduction

Generative adversarial network (GAN) has been the workhorse in deep generative models since its birth for image generation [16], and its popularity arises from the capability of generating clear and realistic images from merely small dimensions. Despite success, the original architecture of GAN only allows for randomly generating images from Gaussian noise, and an important variant of GANs aims to control the generation by pre-defined auxiliary information (e.g., the class labels or texts), constituting the conditional GAN (cGAN). Taking advantages of the auxiliary information, cGANs have been proved to be capable of enhancing the realistic image generation that is conditioned on extra semantic cues [42, 32, 33]. Therefore, the past few years have witnessed the extensive applications of cGANs, including class-conditioned generation [31, 37], style transfer [55], text-to-image translation [42, 51], to name but a few.

Generally speaking, cGANs establish a joint distribution between data $\boldsymbol{\mathcal{X}}$ and auxiliary information $\boldsymbol{\mathcal{Y}}$, i.e., $\{\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}\} \sim p(\mathbf{x}, \mathbf{y})$. Most cGANs agreed on the design of the generator network, in which the auxiliary information is embedded to the input noise [31] or the inter-mediate layers of the generator [11, 37, 9, 50, 40, 4, 33]. As such, the generator aims to sample from the joint distribution $p(\mathbf{x}, \mathbf{y})$. On the other hand, for designing the discriminator, the way we formulate the conditional distribution tells the existing cGANs apart, because $p(\mathbf{x}, \mathbf{y})$ can be formulated by either $p(\mathbf{x}|\mathbf{y})p(\mathbf{y})$ or $p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$. The former calls for transforming the auxiliary information $\boldsymbol{\mathcal{Y}}$ into the discriminator so as to predict $p(\mathbf{x}|\mathbf{y})$, and this can be achieved by concatenating with $\boldsymbol{\mathcal{X}}$ as input [31, 10, 44], or embedding $\boldsymbol{\mathcal{Y}}$ to hidden layers of the discriminator [42, 51]. The latter, however, requires the discriminator to predict the auxiliary information $p(\mathbf{y}|\mathbf{x})$, by for example, additional explicit classifiers [37, 15, 21] or implicit projections [33, 50, 32, 4]. Despite being able to control the generation by pre-defined auxiliary cues, applying cGANs in practice has been significantly restricted owing to their mode collapse [48, 52, 23] and instability [37, 33] problems in training, thus impeding the consistent improvement in the realistic image generation.

Indeed, most discriminators of cGANs build upon the cross-entropy adversarial loss, with an equivalence to the Jensen-Shannon (JS) divergence between generated and real data distributions [2]. Unfortunately, it has been verified both theoretically and empirically that the JS divergence,

*The corresponding author of this paper is Mai Xu, with email: MaiXu@buaa.edu.cn.

which compares two distributions in a "bin-to-bin" manner [25], can easily max out when the two distributions are mis-aligned or supported by low dimensions [2]. Consequently, there exists an issue of gradient vanishing in the discriminator, which misleads the generator to simply learn fixed patterns or completely break down in training [2, 3]. For unconditional generation, this issue has been elegantly addressed by introducing a broad class of distance metrics called integral probability metric (IPM) [35]. Under the umbrella of the theoretical completeness of IPMs, the discriminator operates as certain bounded functions to compare distributions in a "cross-bin" style [25], such that smooth and sufficient gradient can be provided for unconditional generation.

Therefore, it is intuitive to apply IPMs to conditional generation, benefiting from the theoretical completeness of IPMs to stably and consistently improve the generation. However, it is non-trivial to design an IPM-cGAN, due to the non-linear coupling between the data and auxiliary information. In other words, the bounded function of the discriminator prohibits explicitly modelling $p(\mathbf{x}|\mathbf{y})$ or $p(\mathbf{y}|\mathbf{x})$ for conditional generation. Several attempts were proposed to concatenate $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ as an augmented random variable $\widehat{\boldsymbol{\mathcal{X}}}$, and equivalently train the cGAN by an unconditional IPM-GAN [54]. However, it is problematic to straightforwardly combine two random variables at different semantic levels, whereby its deficiency has been proved in many cGANs [28, 33]. Although several cGANs employed certain IPMs, e.g., the Wasserstain distance in their implementations [34, 44, 33], their very basic theories were established upon the cross-entropy form (equivalent to the JS divergence), thus still suffering from the mode collapse and instability problems caused by the "bin-to-bin" comparison. More importantly, the above cGANs are established upon the existence of probability density functions (pdfs) of random variables. This premise, oftentimes taken for granted without verification, may not hold in practice, especially when real-world data such as images and videos essentially reside on low-dimensional manifolds [24, 36].

In this paper, we propose a novel cGAN architecture upon the characteristic function (CF) of random variables, i.e., conditional characteristic function GAN (CCF-GAN). We also noticed several works [1, 30] built upon the CF to achieve enhanced unconditional generation. Those methods, however, by first embedding the data distributions into latent spaces, are problematic in learning joint distributions of the data and auxiliary information in the embedded spaces. In contrast, this paper explicitly establishes the CFs for both generated and real joint distributions. By inspecting that the CF always exists and uniquely corresponds to one distribution, we propose to calculate the difference between CFs as a vehicle to indicate the discrepancy of joint distributions. However, the calculation of CFs requires excessively sampling in the complex domain, which is prohibitive to learn distributions of images that reside in high dimensions. We thus develop the neural network as a proxy to calculate an upper bound of the CF difference, called neural CF (NCF) metric. Based on the NCF, we establish the CCF-GAN by explicitly modelling the conditional distribution from the joint distribution, allowing for a classified treatment on the image and auxiliary information at different semantic levels. Consequently, the superior performances of our CCF-GAN are verified on both synthetic and real-world datasets.

2. Related Work

cGANs basically optimise joint distributions between images and auxiliary information, which fundamentally differ from unconditional GANs that solely optimise image distributions. The joint optimisation of cGANs allows for controllable generation, the key technique in many scenarios including categorical generation and style transfer. Incorporating the auxiliary information within joint distributions has also been proved to further improve the generation quality, of which the existing cGANs are reviewed in the following.

cGANs by conditioning on $p(\mathbf{x}|\mathbf{y})$: The first cGAN [31] proposed to learn the joint distribution by $p(\mathbf{x},\mathbf{y}) = p(\mathbf{y})p(\mathbf{x}|\mathbf{y})$, and concatenated the auxiliary information $\mathcal{Y}$ with the data $\mathcal{X}$ as the input of the generator and discriminator, such that the generation and discrimination processes are both informed by the auxiliary information. Similarly, Laplacian pyramid (LAP) GAN [10] and temporal GAN [44] also concatenated $\mathcal{Y}$ to $\mathcal{X}$ as the input of discriminator, to address the conditional distribution $p(\mathbf{x}|\mathbf{y})$. However, since the data $\mathcal{X}$ and auxiliary information $\mathcal{Y}$ are at different semantic levels, directly concatenating them together may encounter a mismatched information aggregation, leading to instability and inefficiency in training [28, 33]. To relieve this issue, follow-up works [42, 51, 39] proposed to embed $\mathcal{Y}$ to certain hidden layers of the discriminator, such that high-level cues of the data have been extracted and then aggregated by the embedded $\mathcal{Y}$. Unfortunately, the above methods are designed for applying GAN to accomplish specific tasks such as text-to-image translation [42, 51] and image editing [39].

cGANs by conditioning on $p(\mathbf{y}|\mathbf{x})$: Another main trend of cGANs is to decompose $p(\mathbf{x},\mathbf{y})$ into $p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$, whereby $p(\mathbf{y}|\mathbf{x})$ is predicted by either an implicit or explicit classifier. As one of the representative classifier-free methods, the projection-cGAN was proposed to calculate the likelihood ratios and to indicate $p(\mathbf{y}|\mathbf{x})$ by projections, such that the optimisation was implemented under the cross-entropy loss with the theoretical completeness [33]. Due to its simplicity and theoretical beauty, the projection-cGAN has been widely applied in many advanced models, including spectrum normalisation GAN [32], BigGAN [4] and self-attention GAN [50], whereby recent advances including cooperate initialisation [49, 53], knowledge distillation [8] and gradient regularisation [12]. On the other hand, it has been verified that adding a classifier may improve the generation performance [5]. Auxiliary classifier GAN (ACGAN) is one of the most widely employed cGANs with an explicit classifier, which is trained by the marginal distribution and prediction accuracy [37]. However, ACGAN has been criticised by its behaviour of learning biased distributions, which leads to mode collapse especially when training with large amount of auxiliary information [47, 15, 17]. Later improvements therefore include using a twin auxiliary classifier (TAC) in TACGAN [15], training with a contrastive loss in ContraGAN [20], adding an auxiliary discriminative classifier (ADC) in ADCGAN [19] and implementing several regularisations for stable training in ReACGAN [21]. However, all the above cGANs are based on the cross-entropy loss, which suffer from the incomplete comparisons between two well-separated distributions [2] and may result into mode collapse and instability in training.

IPM-cGANs: The IPM has been widely employed for unconditional generation, which successfully reformulates the cross-entropy loss (of predicting real and generated samples) into a theoretically complete distance metric. Notable IPM-GANs include Wasserstein GAN [3], Fisher GAN [34], maximum mean discrepancy GAN [30] and CF-related GANs [1, 30]. To the best of our knowledge, although being extremely potential in addressing the unstable training problems in cGANs, applying IPM to cGANs is still yet to start. This is due to the fact that their IPMs are established based on unconditional generation, and the extension to conditional generation has to concatenate the data and auxiliary information together, such that the unconditional settings can be applied. This, however, significantly limits the power of cGANs because it has been verified that decomposing the joint distribution into marginal and conditional distributions can witness remarkable improvements [28, 33]. We also noticed several cGANs tried to combine cross-entropy prediction and IPMs in an *ad hoc* manner [34, 44, 33], which still suffer from the unstable training.

3. Methodology

3.1. CF Discrepancy

The CF uniquely defines a random variable $\mathcal{V} \in \mathbb{R}^d$ in terms of cumulative density function (cdf) $F_{\mathcal{V}}(\mathbf{v})$, given by

$$\Phi_{\mathcal{V}}(\mathbf{t}) = \mathbb{E}_{\mathcal{V}}[e^{j\mathbf{t}^T\mathbf{v}}] = \int_{\mathbf{v}} e^{j\mathbf{t}^T\mathbf{v}} dF_{\mathcal{V}}(\mathbf{v}), \tag{1}$$

where $\mathbb{E}_{\mathcal{V}}[\cdot]$ denotes the expectation of $\mathcal{V}$. The CF always exists for arbitrary random variables, even when the pdf is not well-defined (for example, the Cantor distribution). When the pdf of a random variable exists, the CF can be formulated as an inverse Fourier transform of $p_{\mathcal{V}}(\mathbf{v})$, i.e., $\Phi_{\mathcal{V}}(\mathbf{t}) = \int_{\mathbf{v}} e^{j\mathbf{t}^T\mathbf{v}} p_{\mathcal{V}}(\mathbf{v}) d\mathbf{v}$. In problems including density estimation and generative modelling, the distribution of random variable $\mathcal{V}$ is typically unknown whilst only a set of independent and identically distributed (i.i.d) samples $\{\mathbf{v}_i\}_{i=1}^n$ from $\mathcal{V}$ is available; this prohibits continuous integral over $F_{\mathcal{V}}(\mathbf{v})$ in CF calculation. Alternatively, we resort to the empirical CF (ECF) that can be calculated as $\bar{\Phi}_{\mathcal{V}}(\mathbf{t}) = \frac{1}{n}\sum_{i=1}^n e^{j\mathbf{t}^T\mathbf{v}_i}$, which is an unbiased and consistent estimator of the population $\Phi_{\mathcal{V}}(\mathbf{t})$ in (1) [13], thus promising a well-defined proxy to approximate the unknown distribution $\mathcal{V}$.

Another appealing property of CF is its boundness, where

$$|\Phi_{\mathcal{V}}(\mathbf{t})| = |\int_{\mathbf{v}} e^{j\mathbf{t}^T\mathbf{v}} dF_{\mathcal{V}}(\mathbf{v})| \leq \int_{\mathbf{v}} |e^{j\mathbf{t}^T\mathbf{v}}| dF_{\mathcal{V}}(\mathbf{v}) = 1, \tag{2}$$

and reaches its maxima at $\Phi_{\mathcal{V}}(\mathbf{0}) = 1$. In other words, two distributions, $\mathcal{V}$ and $\widetilde{\mathcal{V}}$, are automatically aligned in their CFs. It is the fact that comparing two distributions by their pdfs may suffer from misalignment in optimisations, where vanishing gradients and unstable training may exist [2]. This issue motivates the usage of Wasserstein distance, at the cost of increased computational complexity [25] or additional constraints [3]. In contrast, comparing two CFs is naturally resistant to the misalignment issue, whilst enjoying computation ease. We thus use the following l_2-norm discrepancy measurement to compare two distributions (i.e., $\mathcal{V}$ and $\widetilde{\mathcal{V}}$) via their CFs, on the basis of the uniqueness between a random variable and its CF,

$$\mathcal{D}^2_{\mathcal{T}}(\mathcal{V}||\widetilde{\mathcal{V}}) = \int \big(\Phi_{\mathcal{V}}(\mathbf{t}) - \Phi_{\widetilde{\mathcal{V}}}(\mathbf{t})\big)\big(\Phi^*_{\mathcal{V}}(\mathbf{t}) - \Phi^*_{\widetilde{\mathcal{V}}}(\mathbf{t})\big) p_{\mathcal{T}}(\mathbf{t}) d\mathbf{t}. \tag{3}$$

In (3), Φ^* denotes the complex conjugate of Φ, and $p_{\mathcal{T}}(\mathbf{t})$ represents the distribution of $\mathbf{t} \sim \mathcal{T}$ that is able to indicate the discrepancy between $\Phi_{\mathcal{V}}(\mathbf{t})$ and $\Phi_{\widetilde{\mathcal{V}}}(\mathbf{t})$ [30]. It has been proved that when the support of $p_{\mathcal{T}}(\mathbf{t})$ resides in $\mathbb{R}^d$, $\mathcal{D}$ is a valid distance metric to compare two distributions [30]. We may need to point out that besides the l_2 norm, the discrepancy measurement $d(\Phi_{\mathcal{V}}(\mathbf{t}), \Phi_{\widetilde{\mathcal{V}}}(\mathbf{t}))$ can be flexibly chosen by other forms, such as l_1 norm or $\log$ operation.

Furthermore, we focus on the scenario where $\mathcal{V}$ and $\widetilde{\mathcal{V}}$ can be only accessed by their discrete random samples, e.g., $\{\mathbf{v}_i\}_{i=1}^n \sim \mathcal{V}$ for real images and $\{\widetilde{\mathbf{v}}_i\}_{i=1}^{\widetilde{n}} \sim \widetilde{\mathcal{V}}$ for generated images in image generation tasks. Thus, their CFs can be only accessed by the ECFs, which basically falls into the scope of two-sample test problem, and under mild conditions, the equivalence between two ECFs almost surely (a.s.) ensures the equivalence of two distributions with statistical significance [14], thus indicating the consistency between the corresponding two CFs. Due to this equivalence, instead of using the extra notation $\bar{\Phi}_{\mathcal{V}}(\mathbf{t})$, we denote in the sequel the ECF by $\Phi_{\mathcal{V}}(\mathbf{t})$ for simplicity without ambiguity.

More importantly, for conditional generation that involves two joint distributions, for example, $(\mathcal{X}, \mathcal{Y})$ for real images and labels, together with $(\widetilde{\mathcal{X}}, \widetilde{\mathcal{Y}})$ for generated ones, we are able to formulate $\mathcal{V} = (\mathcal{X}, \mathcal{Y})$ and $\widetilde{\mathcal{V}} = (\widetilde{\mathcal{X}}, \widetilde{\mathcal{Y}})$. This way, the above desirable properties including universal existence and uniqueness still hold for their corresponding ECFs, because $\Phi_{\mathcal{X},\mathcal{Y}}(\mathbf{t}) = \Phi_{\mathcal{V}}(\mathbf{t})$ and $\Phi_{\widetilde{\mathcal{X}},\widetilde{\mathcal{Y}}}(\mathbf{t}) = \Phi_{\widetilde{\mathcal{V}}}(\mathbf{t})$. Then, by sampling $\{\mathbf{t}_i\}_{i=1}^{k}$ from $\mathcal{T}$ in (3), we are able to calculate the difference between the two distributions in practice:

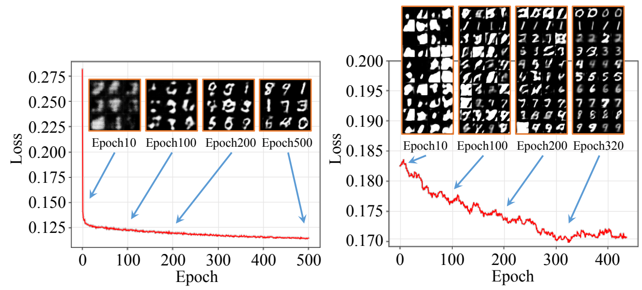
$$\begin{aligned}\mathcal{D}^2_{\mathcal{T}}(\mathcal{V}||\widetilde{\mathcal{V}}) &= \frac{1}{k}\sum_{i=1}^{k}\left(\Phi_{\mathcal{V}}(\mathbf{t}_i) - \Phi_{\widetilde{\mathcal{V}}}(\mathbf{t}_i)\right)\left(\Phi^*_{\mathcal{V}}(\mathbf{t}_i) - \Phi^*_{\widetilde{\mathcal{V}}}(\mathbf{t}_i)\right) \\ &= \frac{1}{k}\sum_{i=1}^{k}\left(\Phi_{\mathcal{X},\mathcal{Y}}(\mathbf{t}_i) - \Phi_{\widetilde{\mathcal{X}},\widetilde{\mathcal{Y}}}(\mathbf{t}_i)\right)\left(\Phi^*_{\mathcal{X},\mathcal{Y}}(\mathbf{t}_i) - \Phi^*_{\widetilde{\mathcal{X}},\widetilde{\mathcal{Y}}}(\mathbf{t}_i)\right) \\ &= \mathcal{D}^2_{\mathcal{T}}(\mathcal{X},\mathcal{Y}||\widetilde{\mathcal{X}},\widetilde{\mathcal{Y}}),\end{aligned} \tag{4}$$

where $\Phi_{\mathcal{X},\mathcal{Y}}(\mathbf{t}_i)$ and $\Phi_{\widetilde{\mathcal{X}},\widetilde{\mathcal{Y}}}(\mathbf{t}_i)$ represent the ECFs for real and generated joint distributions, respectively. It should be pointed out that in (4), the number of samples k plays a crucial role in distinguishing $(\widetilde{\mathcal{X}}, \widetilde{\mathcal{Y}})$ from $(\mathcal{X}, \mathcal{Y})$, so as to indicate sufficient discrepancy for probability estimation. We illustrate in Fig. 1 that without any discriminator modules, optimising a generator network solely by setting $k = 128$ and $\mathcal{T}$ to be the standard Gaussian distribution in (4) can generate roughly realist images towards MNIST digits [27].

However, the grey-scale digital images from MNIST dataset [27] with size 28×28 are simplified scenarios when comparing with real-world images. When optimising images of high dimensions and with diversifying content, k has to increase exponentially, especially for high-dimensional data, encountering the *curse of dimensionality* (*cod*) problem. To address this, rather than Gaussian distribution, $\{\mathbf{t}_i\}_{i=1}^{k}$ need to be smartly chosen. More importantly, in this preliminary experiment of Fig. 1, we straightforwardly concatenated the label information $\mathcal{Y}$ with the images $\mathcal{X}$, which has been verified to be ineffective since the pixel-wise images and class-wise labels are essentially at different semantic levels [28, 33]. In Section 3.2, we first introduce the way of addressing the *cod* problem, followed by a novel way to treat the semantic levels at different importance in Section 3.3.

3.2. Adversarial NCF Learning

To address the *cod* problem when calculating the discrepancy between ECFs, several methods [1, 30], which were born for unconditional generation, proposed to reduce the dimensions of images by learning an embedding function $f(\cdot) : \mathbb{R}^d \to \mathbb{R}^{d'}$, where $d' \leq d$ [1, 30]; this allows for explicitly enumerating $\mathcal{T}$ in the low dimension d' when comparing two embedded distributions $f(\mathcal{V}) \in \mathbb{R}^{d'}$ and $f(\widetilde{\mathcal{V}}) \in \mathbb{R}^{d'}$. However, the embedding requires extra requirements on the function $f(\cdot)$, including injection [1] and bijection [30], resulting into additional hyper-parameters and instability when training GANs. More importantly, the



(a) *w/o.* labels (b) *w.* labels

Figure 1: Preliminary generation results on the MNIST dataset by solely training the generator upon (4). Zero-mean standard Gaussian distribution was chosen as $\mathcal{T}$ in (4), whereby $k = 128$ and output image was of size 28×28.

embedding function $f(\cdot)$ is basically implemented by the discriminator network (also known as the critic), which is highly non-linear. Its extension to conditional generation is therefore highly limited, and the only possible compromise is to embed a concatenated joint distribution $f(\mathcal{V}) = f(\mathcal{X}, \mathcal{Y})$, the way that the majority of IPM-cGANs operate. This compromise, however, is proved to be inefficient [28, 33].

On the other hand, the very basic operation of CFs in (1), namely, $\mathbf{t}^T\mathbf{v}$, projects from high dimension $\mathbb{R}^d$ to a scalar $\mathbb{R}$. Thus, instead of explicitly enumerating $\mathcal{T}$, we propose to implicitly optimise $\mathcal{T}$ when comparing two complicated distributions in (4). This is also motivated by the *Cramer-Wold Theorem* [7], which states that two random variables $\mathcal{V}, \widetilde{\mathcal{V}} \in \mathbb{R}^d$ have the same distribution if and only if distributions of $\mathbf{t}^T\mathcal{V} \in \mathbb{R}$ and $\mathbf{t}^T\widetilde{\mathcal{V}} \in \mathbb{R}$ are the same for all $\mathbf{t} \in \mathbb{R}^d$. In other words, we are able to compare two complicated and high-dimensional distributions, by means of their infinite projections $\mathbb{R}^d \to \mathbb{R}$ in the one-dimensional space. Therefore, instead of explicitly sampling $\mathcal{T}$ from several pre-defined distributions [1, 30], we propose to implicitly search all possible $\mathbf{t}$ and directly output the corresponding projections $\mathbf{t}^T\mathcal{V}$ and $\mathbf{t}^T\widetilde{\mathcal{V}}$, so as to compare the projected distributions in low dimensions. Thus, the difference of two CFs in (4) can be generalised by the projection function f as

$$\mathcal{D}_{\mathcal{F}}(\mathcal{V}||\widetilde{\mathcal{V}}) = \left(\frac{1}{k}\sum_{i=1}^{k}\left(\Phi^{f_i}_{\mathcal{V}} - \Phi^{f_i}_{\widetilde{\mathcal{V}}}\right)\left(\Phi^{f_i*}_{\mathcal{V}} - \Phi^{f_i*}_{\widetilde{\mathcal{V}}}\right)\right)^{\frac{1}{2}}, \tag{5}$$

where $\Phi^{f_i}_{\mathcal{V}}$ is calculated by the i-th projection $f_i(\cdot)$:

$$\Phi^{f_i}_{\mathcal{V}} = \mathbb{E}_{\mathcal{V}}[e^{jf_i(\mathbf{v})}] = \frac{1}{n}\sum_{i_v}^{n} e^{jf_i(\mathbf{v}_{i_v})}. \tag{6}$$

In (5), $f_i(\mathbf{v}_{i_v})$ is parameterised by the proposed NCF network, whereby the input is $\mathbf{v}_{i_v}$ and $f_i(\mathbf{v}_{i_v})$ represents the i-th dimension output of the NCF network.

Furthermore, compared with excessively sampling by varying $f(\mathbf{v})$, it is more efficient to decide the "best repre-

Algorithm 1: Training algorithm of the proposed CCF-GAN.

input: Real images $\mathcal{P}_r$; standard Gaussian distribution $\mathcal{P}_\mathcal{N}$; number of class labels c; category distribution of c classes $\mathcal{P}_\mathcal{C}$; batch size b_s; learning rate l_r; training steps s_d and s_g for discriminator and generator per iteration
output: Net parameters θ_d and θ_g for the discriminator $f(\cdot)$ and generator $g(\cdot)$, respectively
while *θ_d and θ_g not converged* **do**
    /* train discriminator */
    for *index* $\leftarrow 1$ *to* s_d **do**
        Sample from real distribution:
        $\{\mathbf{x}_j, \mathbf{y}_j\}_{j=1}^{b_s} \sim \mathcal{P}_r$; $\{\mathbf{z}_j\}_{j=1}^{b_s} \sim \mathcal{P}_\mathcal{N}$; $\{\widetilde{\mathbf{y}}_j\}_{j=1}^{b_s} \sim \mathcal{P}_\mathcal{C}$; $\{\mathbf{t}_y^i\}_{i=1}^{k}$ sampled by the linear space rule from $[-1, 1]$
        Calculate NCF of real images $\mathcal{X}$ and generated images $\widetilde{\mathcal{X}} = g(\mathcal{Z})$:
        $\{f_i(\mathbf{x}_j)\}_{i=1,j=1}^{k,b_s} \leftarrow \{\mathbf{x}_j\}_{j=1}^{b_s}$; $\{f_i(\widetilde{\mathbf{x}}_j)\}_{i=1,j=1}^{k,b_s} = \{f_i\big(g(\mathbf{z}_j)\big)\}_{i=1,j=1}^{k,b_s} \leftarrow \{g(\mathbf{z}_j)\}_{j=1}^{b_s}$
        Calculate ECFs $\Phi_{\mathcal{X},\mathcal{Y}}^{f_i}(\mathbf{t}_y^i)$ and $\Phi_{\widetilde{\mathcal{X}},\widetilde{\mathcal{Y}}}^{f_i}(\mathbf{t}_y^i)$ by (11), whereby $p(\mathbf{y}_{i_y}|\mathbf{x}_{i_x})$ (or $p(\widetilde{\mathbf{y}}_{i_y}|\widetilde{\mathbf{x}}_{i_x})$) is ground-truth or predicted by [15]
        Calculate discriminator loss by (12): $\mathcal{L}_\mathcal{D} = -\mathcal{L}(\mathcal{X}, \mathcal{Y}||\widetilde{\mathcal{X}}, \widetilde{\mathcal{Y}})$
        Update: $\theta_d \leftarrow \theta_d + l_r \cdot \text{Adam}\,(\theta_d, \nabla_{\theta_d}[\mathcal{L}_\mathcal{D}])$
    /* train generator */
    for *index* $\leftarrow 1$ *to* s_g **do**
        Sample from real distribution:
        $\{\mathbf{x}_j, \mathbf{y}_j\}_{j=1}^{b_s} \sim \mathcal{P}_r$; $\{\mathbf{z}_j\}_{j=1}^{b_s} \sim \mathcal{P}_\mathcal{N}$; $\{\widetilde{\mathbf{y}}_j\}_{j=1}^{b_s} \sim \mathcal{P}_\mathcal{C}$; $\{\mathbf{t}_y^i\}_{i=1}^{k}$ sampled by the linear space rule from $[-1, 1]$
        Calculate NCF of real images $\mathcal{X}$ and generated images $\widetilde{\mathcal{X}} = g(\mathcal{Z})$:
        $\{f_i(\mathbf{x}_j)\}_{i=1,j=1}^{k,b_s} \leftarrow \{\mathbf{x}_j\}_{j=1}^{b_s}$; $\{f_i(\widetilde{\mathbf{x}}_j)\}_{i=1,j=1}^{k,b_s} = \{f_i\big(g(\mathbf{z}_j)\big)\}_{i=1,j=1}^{k,b_s} \leftarrow \{g(\mathbf{z}_j)\}_{j=1}^{b_s}$
        Calculate ECFs $\Phi_{\mathcal{X},\mathcal{Y}}^{f_i}(\mathbf{t}_y^i)$ and $\Phi_{\widetilde{\mathcal{X}},\widetilde{\mathcal{Y}}}^{f_i}(\mathbf{t}_y^i)$ by (11), whereby $p(\mathbf{y}_{i_y}|\mathbf{x}_{i_x})$ (or $p(\widetilde{\mathbf{y}}_{i_y}|\widetilde{\mathbf{x}}_{i_x})$) is ground-truth or predicted by [15]
        Calculate generator loss by (12): $\mathcal{L}_\mathcal{G} = \mathcal{L}(\mathcal{X}, \mathcal{Y}||\widetilde{\mathcal{X}}, \widetilde{\mathcal{Y}})$
        Update: $\theta_g \leftarrow \theta_g + l_r \cdot \text{Adam}\,\big(\theta_g, \nabla_{\theta_g}[\mathcal{L}_\mathcal{G}]\big)$;

sentative" samples that are able to maximally distinguish the two CFs in $\mathcal{D}_\mathcal{F}(\mathcal{V}||\widetilde{\mathcal{V}})$ in (5), as follows

$$\mathcal{L}(\mathcal{V}||\widetilde{\mathcal{V}}) = \max_f \mathcal{D}_\mathcal{F}(\mathcal{V}||\widetilde{\mathcal{V}}). \tag{7}$$

Lemma 1. *For any two random variables $\mathcal{V}, \widetilde{\mathcal{V}} \in \mathbb{R}^d$, $\mathcal{L}(\mathcal{V}||\widetilde{\mathcal{V}}) \geq \mathcal{D}_\mathcal{T}(\mathcal{V}||\widetilde{\mathcal{V}})$ for any $\mathcal{T}$, where $\mathcal{D}_\mathcal{T}(\mathcal{V}||\widetilde{\mathcal{V}})$ is defined in* (4).

Lemma 1[1] proves an upper bound of $\mathcal{L}(\mathcal{V}||\widetilde{\mathcal{V}})$ against the true CF discrepancy $\mathcal{D}_\mathcal{T}(\mathcal{V}||\widetilde{\mathcal{V}})$. This way, minimising $\mathcal{L}(\mathcal{V}||\widetilde{\mathcal{V}})$ naturally reduces the difference between two distributions, as measured by $\mathcal{D}_\mathcal{T}(\mathcal{V}||\widetilde{\mathcal{V}})$. We further provide in Lemma 2 that the measurement $\mathcal{L}(\mathcal{V}||\widetilde{\mathcal{V}})$ is a valid distance metric, which is able to precisely reflect the difference between two distributions.

Lemma 2. *If $\mathcal{V}, \widetilde{\mathcal{V}} \in \mathbb{R}^d$ are two random variables, $\mathcal{L}(\mathcal{V}||\widetilde{\mathcal{V}})$ in* (7) *is a valid distance metric.*

3.3. Conditional Generation by CCF-GAN

By far, the conditional generation can be achieved by setting $\mathcal{V} = (\mathcal{X}, \mathcal{Y})$ as illustrated by Fig. 1. However, since the image $\mathcal{X}$ and auxiliary information $\mathcal{Y}$ reside at different semantic levels, directly stacking them together is problematic in cGANs [28, 33]. In this section, we propose to treat them separately such that the auxiliary information $\mathcal{Y}$ can be well accommodated along with generating $\mathcal{X}$, thus enjoying improved generation performances. More specifically, the definition of CF allows for an explicit decomposition on $p(\mathbf{x})$ and $p(\mathbf{y}|\mathbf{x})$ as follows,

$$\begin{aligned}\Phi_\mathcal{V}(\mathbf{t}) = \Phi_{\mathcal{X},\mathcal{Y}}(\mathbf{t}) &= \int_\mathbf{x}\int_\mathbf{y} e^{j(\mathbf{t}_x^T\mathbf{x}+\mathbf{t}_y^T\mathbf{y})} p(\mathbf{x},\mathbf{y}) d\mathbf{x} d\mathbf{y} \\ &= \int_\mathbf{x}\Big[\int_\mathbf{y} e^{j\mathbf{t}_y^T\mathbf{y}} p(\mathbf{y}|\mathbf{x}) d\mathbf{y}\Big] e^{j\mathbf{t}_x^T\mathbf{x}} p(\mathbf{x}) d\mathbf{x},\end{aligned} \tag{8}$$

where $\mathbf{t} = [\mathbf{t}_x^T, \mathbf{t}_y^T]^T$. We may need to point out that (8) plays a key role in our CCF-GAN, which effectively decomposes $\mathcal{Y}$ from $\mathcal{X}$. In many tasks, the auxiliary information follows the discretely distribution, e.g., the class labels. Thus, we are able to obtain the CF of $p(\mathbf{y}|\mathbf{x})$ in (8) as

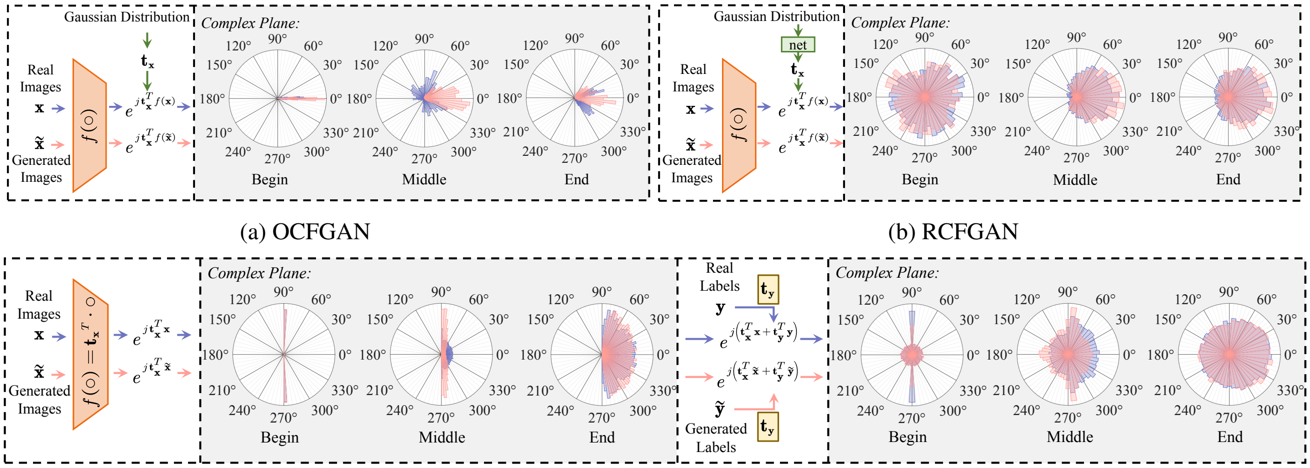
$$\int_\mathbf{y} e^{j\mathbf{t}_y^T\mathbf{y}} p(\mathbf{y}|\mathbf{x}) d\mathbf{y} = \sum_{i=1}^{c} e^{j\mathbf{t}_y^T\mathbf{y}_i} p(\mathbf{y}_i|\mathbf{x}), \tag{9}$$

where c is the number of discrete values of $\mathcal{Y}$. Correspondingly, the ECF of $(\mathcal{X}, \mathcal{Y})$ now arrives at

$$\Phi_{\mathcal{X},\mathcal{Y}}(\mathbf{t}) = \frac{1}{n}\sum_{i_x=1}^{n}\sum_{i_y=1}^{c} e^{j\mathbf{t}_y^T\mathbf{y}_{i_y}} p(\mathbf{y}_{i_y}|\mathbf{x}_{i_x}) e^{j\mathbf{t}_x^T\mathbf{x}_{i_x}}. \tag{10}$$

More importantly, the image distribution $\mathcal{X}$ typically resides in high dimensions and thus requires smart strategies to avoid

[1]Please refer to the supplementary material for the proofs of all lemmas.



(a) OCFGAN (b) RCFGAN

(c) CCF-GAN

Figure 2: Evolution of adversarial learning when training CF-related GANs, in terms of projected generated/real samples $e^{j\mathbf{t}^T\mathbf{v}}$ on the complex plane of CFs, at the beginning, middle and end stages. The output $e^{j\mathbf{t}^T\mathbf{v}}$ from the discriminator was averaged in a batch-size manner for visualisation ease, which is then plotted by angle-based histograms on the complex plane. Pink color indicates the histogram for generated samples while purple indicates that for the real samples. Please note that (a) OCFGAN [1] and (b) RCFGAN [29] were proposed for unconditional generation solely, whereby $e^{j\mathbf{t}^T\mathbf{x}}$ is plotted and analysed. (c) The proposed CCF-GAN is analysed by illustrating the dynamics of both $e^{j\mathbf{t}^T\mathbf{x}}$ and $e^{j\mathbf{t}^T\mathbf{y}}$ for conditional generation.

cod when choosing $\mathbf{t}_x$. In contrast, the auxiliary information $\mathcal{Y}$ is with relatively low dimensions. Therefore, instead of directly enumerating $\mathbf{t}_x^T\mathcal{X}$ in high dimensions, the NCF network $f(\cdot)$ proposed in Section 3.2 is employed to calculate $\mathbf{t}_x^T\mathcal{X}$ in (10), which now becomes

$$\Phi^f_{\mathcal{X},\mathcal{Y}}(\mathbf{t}_y) = \frac{1}{n}\sum_{i_x=1}^{n}\sum_{i_y=1}^{c} e^{j\mathbf{t}_y^T\mathbf{y}_{i_y}} p(\mathbf{y}_{i_y}|\mathbf{x}_{i_x}) e^{jf(\mathbf{x}_{i_x})}. \quad (11)$$

Recall that we use superscript f to indicate the ECF is calculated by the proposed NCF network. In (11), $p(\mathbf{y}_{i_y}|\mathbf{x}_{i_x})$ can be directly chosen by the ground-truth labels to achieve the conditional generation, although the performance can be further improved by training an additional classifier [15]; this shall be analysed in our ablation study.

Therefore, by substituting (11) into (5) and (7), we arrive at the final loss function of training our CCF-GAN:

$$\min_g \mathcal{L}(\mathcal{X},\mathcal{Y}||\widetilde{\mathcal{X}},\widetilde{\mathcal{Y}}) = \min_g \max_f \mathcal{D}_{\mathcal{F}}(\mathcal{X},\mathcal{Y}||\widetilde{\mathcal{X}},\widetilde{\mathcal{Y}})$$
$$= \left(\frac{1}{k}\sum_{i=1}^{k}\left(\Phi^{f_i}_{\mathcal{X},\mathcal{Y}}(\mathbf{t}_y^i) - \Phi^{f_i}_{\widetilde{\mathcal{X}},\widetilde{\mathcal{Y}}}(\mathbf{t}_y^i)\right)\left(\Phi^{f_i*}_{\mathcal{X},\mathcal{Y}}(\mathbf{t}_y^i) - \Phi^{f_i*}_{\widetilde{\mathcal{X}},\widetilde{\mathcal{Y}}}(\mathbf{t}_y^i)\right)\right)^{\frac{1}{2}} \quad (12)$$

In (12), $\mathbf{t}_y^i$ represents the i-th sample of $\mathbf{t}_y$. Since $\mathcal{Y}$ resides in low dimensions, we are able to sample $\mathbf{t}_y$ with fixed rules, thus reducing the complexity when training the proposed CCF-GAN. Recall that $\widetilde{\mathcal{X}}$ denotes the generated images from the generator network $g(\cdot)$, and $f(\cdot)$ is the proposed NCF network, which acts as the discriminator in our CCF-GAN.

To the best of our knowledge, the IPMs have been incorporated in existing cGANs by non-linear transform functions, which make the decomposition between $\mathcal{X}$ and $\mathcal{Y}$ intractable. In contrast, our CCF-GAN, benefiting from proposing the NCF network to directly output $\mathbf{t}_x^T\mathbf{x}$, is able to explicitly extract $\mathcal{X}$ from the joint distribution, and the remaining part is formulated by the conditional distribution $p(\mathbf{y}|\mathbf{x})$. This way, the data distribution and auxiliary information can be well learned with different importance, allowing for an optimised discrepancy measure between the real $(\mathcal{X},\mathcal{Y})$ and generated $(\widetilde{\mathcal{X}},\widetilde{\mathcal{Y}})$ joint distributions. In practice, we implement the proposed NCF network as the discriminator, so as to measure the generated distribution $(\widetilde{\mathcal{X}},\widetilde{\mathcal{Y}})$ from the generator $g(\cdot)$. The details are presented in Algorithm 1, whereas the pipeline is provided in Fig. 2. We further illustrate the superiority of the proposed CCF-GAN against existing CF related GANs in Fig. 2. As can be seen from this figure, CCF-GAN can well separate real and generated samples at the middle stage, whereby RCFGAN fails. At the end stage, real and generated samples are aligned by CCF-GAN, whereas separation still exists in OCFGAN.

4. Experiment

4.1. Experimental Settings

Datasets: By comparing the proposed CCF-GAN with other state-of-the-art cGANs, we performed the experiments to evaluate the performances of conditional generation on 1 synthetic dataset and 3 widely accepted real-world datasets, namely, CIFAR10 [26], VGGFace2 [38] and ImageNet [43]. For the synthetic dataset, we employed a mixture of 3 von Mises–Fisher (vMF) distributions [6], and their parameters

$\{p, \tau, \theta\}$ were set to $\{0.33, 30, {}^{2\pi}/_{3}\}$, $\{0.33, 30, {}^{4\pi}/_{3}\}$ and $\{0.33, 30, 2\pi\}$, respectively, where $100k$ points were randomly sampled. The real samples were plotted in Fig. 3-(e). More importantly, the consideration of using vMF clusters is because the vMF distribution is basically supported in low dimensions, which can effectively mimic the real-world scenarios where the data are typically in high dimensions and the generating spaces reside on low dimensions. For real-world scenarios, images in CIFAR10 dataset were of size 32×32. We followed [15] to randomly select 200, 500 and $1,000$ classes from the VGGFace2 dataset, denoted as VGGFace_c200, VGGFace_c500, and VGGFace_c1000. Then, the images were centercropped and resized to 64×64. For ImageNet, we resized the images to resolution of 128×128.

Metrics: The widely applied Fréchet inception distance (FID) [18] metric was adpoted in our evaluation to assess the generation quality of GANs, which basically implements the Wasserstein distance between the real and generated features extracted from the Inception_V3 network. We also adopted the Inception Score (IS) [46] to evaluate the conditional generation performances. When calculating FID and IS scores, we sampled $50k$ images for both generated and real images, which is a common choice in many reported GAN results. Furthermore, we calculated the precision and recall metrics [45], to indicate the mode collapse problem in generation. The stability was also evaluated by repeatedly training GANs under various conditions.

Baselines: We compared the proposed CCF-GAN with the BigGAN [4], ACGAN [37], TACGAN [15] and ADCGAN [19]. Besides, FisherGAN [34] and cRCFGAN[2] [29] were adopted for comparison as conditional IPM-GANs. Furthermore, we also evaluated our CCF-GAN with several most recent GANs without the classifier, including CoopInit [53], KD-DLGAN [8] and DigGAN [12]. We implemented our CCF-GAN on the Pytorch BigGAN platform[3], by using exactly the same architecture for the generator and discriminator networks as the BigGAN [4]. All the comparing cGANs were trained and tested based on the Pytorch BigGAN platform, under the same architecture. Most recently, there comes with a new rising-star platform called the StudioGAN[4] [22], which implements ContraGAN [20] and ReACGAN [21]. Because the new StudioGAN employed random flipping and different image resize functions by default, we believe it is unfair to report the result upon the StudioGAN platform. Otherwise, it might be unclear to show the origin of our improvements. Indeed, we have witnessed further improvements on the StudioGAN platform on all the datasets, which we decided to put in the supplementary material. Our code, implemented on both the BigGAN and StudioGAN platforms, is available at `https://github.com/Zhangjialu126/ccf_gan`.

[2]Due to non-linear coupling, RCFGAN was designed for unconditional generation. cRCFGAN is its compromised variant by channel-wise concatenating images and labels as augmented input for condition generation.

[3]`https://github.com/ajbrock/BigGAN-PyTorch`

[4]`https://github.com/POSTECH-CVLab/PyTorch-StudioGAN`

Table 1: Comparison on FID scores against existing state-of-the-art methods. Symbol * denotes that the results are reported from the corresponding paper, whereas † from [15]. Otherwise, we ran the available codes by the corresponding default settings. We denote the best FID by red color and the second best by blue color.

Method	CIFAR10	VGGFace_c200	VGGFace_c500	VGGFace_c1000
BigGAN [4]	14.73*	66.23†	43.10†	24.07†
ACGAN [37]	8.01	95.70†	31.90†	—
FisherGAN [34]	11.46	13.28	9.02	7.30
TACGAN [15]	8.42	29.12†	12.42†	13.60†
cRCFGAN [29]	6.90	27.03	18.03	20.72
ContraGAN [20]	10.60*	—	—	—
ReACGAN [21]	6.22	13.48	7.19	6.47
ADCGAN [19]	7.17	18.64	11.34	7.94
DigGAN [12]	8.49*	—	—	—
CoopInit$_{BigGAN}$ [53]	6.95*	—	—	—
KD-DLGAN [8]	8.19*	—	—	—
CCF-GAN (Ours)	**6.08**	**11.61**	**6.81**	**5.70**

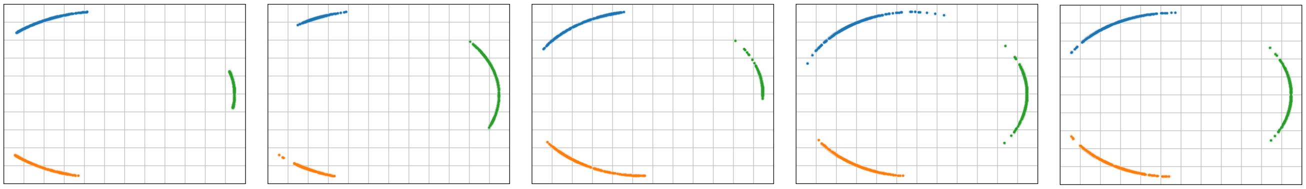
Technical details: In our experiments, we selected a steady learning rate of 0.0001 for generator and 0.0002 for discriminator with classifier. Although being able to achieve conditional generation by directly inputting ground-truth labels, the default setting of our CCF-GAN included the classifier [15]. The discriminator, together with the classifier, was trained 2 steps per generator update. For other comparing methods that were replicated by their public repositories, we set the same hyper-parameters as those in the corresponding papers. More importantly, the batch sizes for CIFAR10, VGGFace_c200, VGGFace_c500 and VGGFace_c1000 were set to 64. Batch size for ImageNet was set to 256.

4.2. Distribution Fitting Results on Synthetic Data

We illustrate in Fig. 3 the comparisons among the ACGAN, TACGAN, ADCGAN and the proposed CCF-GAN, on the 2D synthetic dataset. As can be seen from this figure, our CCF-GAN almost recovered the ground-truth distribution, whereas others are either over-concentrated (ADCGAN) or imbalanced (ACGAN and TACGAN). This validates the effectiveness of employing the NCF in our CCF-GAN, which stably and accurately measured two distributions even when they were supported in low dimensions. In contrast, existing cGANs that are designed based on the cross-entropy loss may suffer from the ill-posed discrepancy measure, such that the fitted distributions were biased.

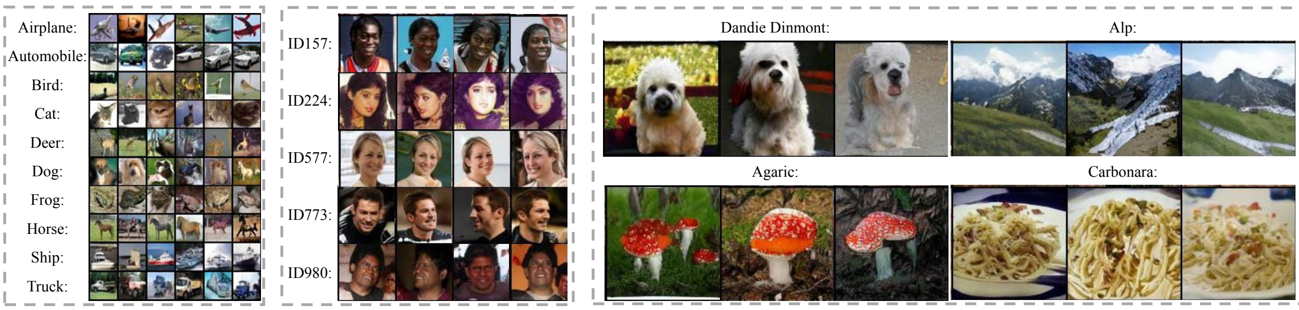
4.3. Realistic Image Generation Results

We also compared our CCF-GAN with existing state-of-the-art baselines in Tables 1 and 2. As can be seen from Table 1, the proposed CCF-GAN achieved the lowest (best)



(a) ACGAN (b) TACGAN (c) ADCGAN (d) CCF-GAN (Ours) (e) Ground truth

Figure 3: Distribution fitting results on 2D synthetic dataset, which consists of $100k$ samples from the mixture of vMF distributions. Please note that ACGAN, TACGAN, ADCGAN and CCF-GAN were trained by the same networks, which consist of 4-layer (for generator) and 3-layer (for discriminator) fully connected neural networks of hidden size equal to 10.



(a) CIFAR10 (b) VGGFace_c1000 (c) ImageNet

Figure 4: Conditional image generation on CIFAR10, VGGFace2_c1000 and ImageNet datasets by the proposed CCF-GAN. Each row represents one class-conditioned generation.

Table 2: Comparison on the ImageNet. Symbol * denotes that the results are reported from [19], whereas † from [15], ‡ from [21] and ** for [20]. Otherwise, we ran the available codes by the corresponding default settings.

Method	ImageNet	
	FID	IS
BigGAN [4]	22.77†	38.05†
ContraGAN [20]	19.69**	31.10**
ACGAN [37]	184.41†	7.26†
TACGAN [15]	23.75†	28.86†
ReACGAN [21]	13.98‡	68.27‡
ADCGAN [19]	16.75*	55.43*
CCF-GAN (Ours)	**11.34**	**180.84**

FID against all the compared methods. Similar results can be also concluded in Table 2, whereby the proposed CCF-GAN achieved the value 11.34 of FID by training under the batch size of 256 for the ImageNet dataset. The IS score of our CCF-GAN, however, was much remarkable and reached 180.84, almost tripled against the second best ReACGAN.

We further present in Fig. 4 the conditional generation results of our CCF-GAN. As can be seen from this figure, our CCF-GAN achieved high-quality image generation. More importantly, by inspecting each row, the class-wise semantics are obvious and the generated images within each class are of diversifying content, which verifies that the proposed CCF-GAN, by incorporating the CF distance measure, is able to overcome the mode collapse issue. In Fig. 5, the interpolation was performed across different classes, whereby the interpolation between two different faces is smooth, verifying the desirable continuity of the latent space learnt by our CCF-GAN. Qualitative comparisons, together with more subjective results and analysis, are provided in the supplementary material.

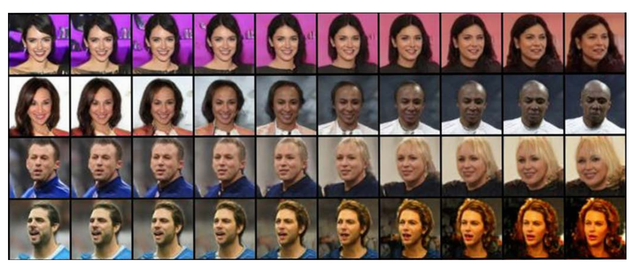
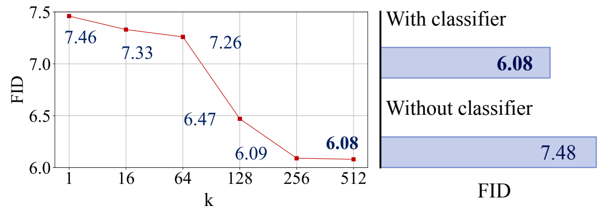


Figure 5: Interpolation across class labels of the proposed CCF-GAN, which was trained on VGGFace_c1000 dataset.

4.4. In-depth Analysis

Ablation study on k, classifier and $\mathbf{t}_y$: Since the number of $\mathbf{t}$ samples, namely, k, plays a crucial role in distinguishing CFs between generated and real distributions, FIDs of varying k are plotted in Fig. 6-(a). We thus can conclude that in complicated real-world scenarios, the proposed NCF effectively resolves *cod* issue, and $k = 256$ is sufficient to be the best among the existing baselines. Another ablation



(a) FIDs with different k (b) FIDs *w./wo.* classifier

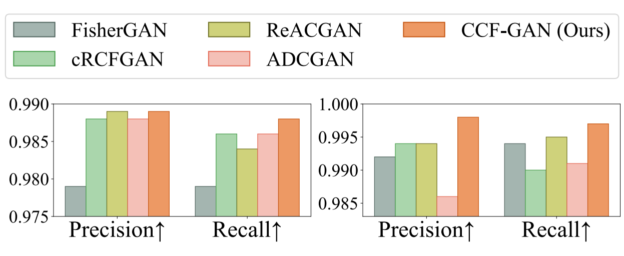
Figure 6: Ablation study on CIFAR dataset. (a) Training CCF-GAN by varying k. (b) Training CCF-GAN with and without the classifier.

Table 3: Ablations on different choices on $\mathbf{t}_y$.

	Fixed $\mathbf{t}_y$			Uniform $\mathbf{t}_y$	Gaussian $\mathbf{t}_y$
α	0.5	**1**	10	—	—
FID↓	6.51	**6.08**	7.02	7.05	7.07

investigates the usage of classifier, as shown in Fig. 6-(b). As can be seen from this figure, our CCF-GAN can still achieve conditional generation by directly using the ground-truth labels as $p(\mathbf{y}_{i_y}|\mathbf{x}_{i_x})$ in (11), i.e., without the classifier. However, training a classifier witnessed improvements on FIDs in our CCF-GAN, which is in accordance with [5]. Moreover, we also ablated on different choices on $\mathbf{t}_y$, including fixed linear space rule of range $[-\alpha, \alpha]$, as well as random samples from uniform and Gaussian distributions. Table 3 indicates that the proposed CCF-GAN performs well under $\mathbf{t}_y$ from different distributions, particularly when $\mathbf{t}_y$ was fixed to be $[-1, 1]$.

Analysis on mode collapse: We quantitatively evaluated the mode collapse of generation by the precision and recall metrics [45] plotted in Fig. 7. From this figure, we can find that our CCF-GAN achieved the highest (best) precision and recall values, and the improvements on the recall were even more significant, which verifies the capability of relieving mode collapse of our CCF-GAN.



(a) CIFAR10 (b) VGGFace_c1000

Figure 7: Precision and recall metrics [45] on CIFAR10 and VGGFace_c1000 datasets.

Improvements on training stability: We further evaluated the stability by repeatedly training cGANs under different conditions. For fair comparisons, we disabled the exponential moving average module and set the discriminator training step to 1 for all methods. We varied 2 learning rates {0.0001, 0.001}, 3 batch sizes {32,64,128}, 2 base channel numbers {64,96}, 2 generator architectures {simple convolution layers [41], residual blocks [4]} and 2 discriminator architectures {simple convolution layers [41], residual blocks [4]}, thus obtaining 48 different challenging conditions. The other parameters of cGANs were kept by their corresponding default values. For each method, 48 trials were implemented to train the model by $100k$ iterations, corresponding to the 48 conditions. The best FID of each trial was then recorded. Fig. 8 reports the number of successful trails filtered by different FID thresholds for our and other state-of-the-art methods. Note that a trail is defined to be successful when its best FID is lower than the threshold. Therefore, it is obvious that our CCF-GAN consistently achieved the most numbers of successful trials given all FID thresholds. The significant improvement on the training stability verifies the theoretical completeness and benefits of our CCF-GAN.

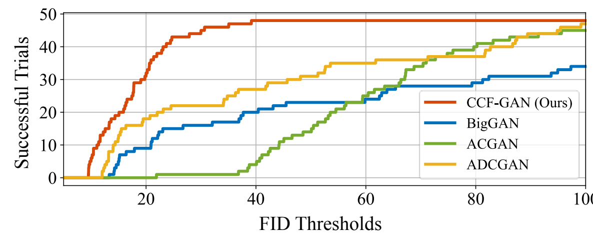


Figure 8: Stability evaluation under 48 conditions. The horizontal axis represents FID thresholds, and vertical axis denotes the number of trials whose best FIDs are lower than the corresponding FID threshold within $100k$ iterations.

5. Conclusion

In this paper, we have proposed a novel CCF-GAN for consistently improving the conditional generation performances on both synthetic and real-world datasets. Different from the existing cGANs built upon the cross-entropy loss, our CCF-GAN benefits from the characteristic function (CF), which processes unique and universal correspondence to a random variable, even when the random variable does not possess probability density function. On the basis of the CF, we have proposed an efficient neural characteristic function (NCF) network to calculate the difference between CFs with theoretical completeness. We further explicitly decomposed the joint distribution by the marginal and conditional distributions, with classified treatment for different semantics levels. This way, CCF-GAN has overcome the deficiency of almost all cGANs of employing the cross-entropy loss. The experimental results have verified that the proposed CCF-GAN achieved the best conditional generation, whilst significantly reducing mode collapse and unstablility in cGANs.

Acknowledgments. This work was supported by NSFC under Grants 62206011, 62250001 and 62231002, and Beijing Natural Science Foundation under Grant JQ20020 and L223021.

References

[1] Abdul Fatir Ansari, Jonathan Scarlett, and Harold Soh. A characteristic function approach to deep implicit generative modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7478–7487, 2020.

[2] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large-scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[5] Si-An Chen, Chun-Liang Li, and Hsuan-Tien Lin. A unified view of cGANs with and without classifiers. *Advances in Neural Information Processing Systems*, 34:27566–27579, 2021.

[6] Mário Costa, Visa Koivunen, and H Vincent Poor. Estimating directional statistics using wavefield modeling and mixtures of von-mises distributions. *IEEE Signal Processing Letters*, 21(12):1496–1500, 2014.

[7] Harald Cramér and Herman Wold. Some theorems on distribution functions. *Journal of the London Mathematical Society*, 1(4):290–294, 1936.

[8] Kaiwen Cui, Yingchen Yu, Fangneng Zhan, Shengcai Liao, Shijian Lu, and Eric P Xing. Kd-dlgan: Data limited image generation via knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3872–3882, 2023.

[9] Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. *Advances in Neural Information Processing Systems*, 30, 2017.

[10] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. *Advances in Neural Information Processing Systems*, 28, 2015.

[11] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016.

[12] Tiantian Fang, Ruoyu Sun, and Alex Schwing. Diggan: Discriminator gradient gap regularization for gan training with limited data. *Advances in Neural Information Processing Systems*, 35:31782–31795, 2022.

[13] Andrey Feuerverger and Roman A Mureika. The empirical characteristic function and its applications. *The annals of Statistics*, pages 88–97, 1977.

[14] Sebastian J Goerg and Johannes Kaiser. Nonparametric testing of distributions—the epps–singleton two-sample test using the empirical characteristic function. *The Stata Journal*, 9(3):454–465, 2009.

[15] Mingming Gong, Yanwu Xu, Chunyuan Li, Kun Zhang, and Kayhan Batmanghelich. Twin auxilary classifiers gan. *Advances in neural information processing systems*, 32, 2019.

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[17] Ligong Han, Anastasis Stathopoulos, Tao Xue, and Dimitris Metaxas. Unbiased auxiliary classifier gans with mine. *arXiv preprint arXiv:2006.07567*, 2020.

[18] 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. *Advances in neural information processing systems*, 30, 2017.

[19] Liang Hou, Qi Cao, Huawei Shen, Siyuan Pan, Xiaoshuang Li, and Xueqi Cheng. Conditional gans with auxiliary discriminative classifier. In *International Conference on Machine Learning*, pages 8888–8902. PMLR, 2022.

[20] Minguk Kang and Jaesik Park. Contragan: Contrastive learning for conditional image generation. *Advances in Neural Information Processing Systems*, 33:21357–21369, 2020.

[21] Minguk Kang, Woohyeon Shim, Minsu Cho, and Jaesik Park. Rebooting acgan: Auxiliary classifier gans with stable training. *Advances in Neural Information Processing Systems*, 34:23505–23518, 2021.

[22] MinGuk Kang, Joonghyuk Shin, and Jaesik Park. StudioGAN: A Taxonomy and Benchmark of GANs for Image Synthesis. *2206.09479 (arXiv)*, 2022.

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

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[25] Soheil Kolouri, Se Rim Park, Matthew Thorpe, Dejan Slepcev, and Gustavo K Rohde. Optimal mass transport: Signal processing and machine-learning applications. *IEEE signal processing magazine*, 34(4):43–59, 2017.

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

[27] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[28] Chunyuan Li, Hao Liu, Changyou Chen, Yuchen Pu, Liqun Chen, Ricardo Henao, and Lawrence Carin. Alice: Towards understanding adversarial learning for joint distribution matching. *Advances in neural information processing systems*, 30, 2017.

[29] Shengxi Li, Zeyang Yu, Min Xiang, and Danilo Mandic. Reciprocal adversarial learning via characteristic functions. *Advances in Neural Information Processing Systems*, 33:217–228, 2020.

[30] Shengxi Li, Zeyang Yu, Min Xiang, and Danilo Mandic. Reciprocal gan through characteristic functions (rcf-gan). *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (01):1–1, 2022.

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

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

[33] Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. *arXiv preprint arXiv:1802.05637*, 2018.

[34] Youssef Mroueh and Tom Sercu. Fisher gan. *Advances in Neural Information Processing Systems*, 30, 2017.

[35] Alfred Müller. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29(2):429–443, 1997.

[36] Hariharan Narayanan and Sanjoy Mitter. Sample complexity of testing the manifold hypothesis. *Advances in neural information processing systems*, 23, 2010.

[37] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*, pages 2642–2651. PMLR, 2017.

[38] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015.

[39] Guim Perarnau, Joost Van De Weijer, Bogdan Raducanu, and Jose M Álvarez. Invertible conditional gans for image editing. *arXiv preprint arXiv:1611.06355*, 2016.

[40] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[41] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[42] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International conference on machine learning*, pages 1060–1069. PMLR, 2016.

[43] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[44] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *Proceedings of the IEEE international conference on computer vision*, pages 2830–2839, 2017.

[45] Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. *Advances in neural information processing systems*, 31, 2018.

[46] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[47] Rui Shu, Hung Bui, and Stefano Ermon. Ac-gan learns a biased distribution. In *NIPS Workshop on Bayesian Deep Learning*, volume 8, 2017.

[48] Yan Wu, Jeff Donahue, David Balduzzi, Karen Simonyan, and Timothy Lillicrap. Logan: Latent optimisation for generative adversarial networks. *arXiv preprint arXiv:1912.00953*, 2019.

[49] Jianwen Xie, Yang Lu, Ruiqi Gao, Song-Chun Zhu, and Ying Nian Wu. Cooperative training of descriptor and generator networks. *IEEE transactions on pattern analysis and machine intelligence*, 42(1):27–45, 2018.

[50] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International conference on machine learning*, pages 7354–7363. PMLR, 2019.

[51] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915, 2017.

[52] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. *Advances in Neural Information Processing Systems*, 33:7559–7570, 2020.

[53] Yang Zhao, Jianwen Xie, and Ping Li. CoopInit: Initializing generative adversarial networks via cooperative learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9):11345–11353, 2023.

[54] Ming Zheng, Tong Li, Rui Zhu, Yahui Tang, Mingjing Tang, Leilei Lin, and Zifei Ma. Conditional wasserstein generative adversarial network-gradient penalty-based approach to alleviating imbalanced data classification. *Information Sciences*, 512:1009–1023, 2020.

[55] 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*, pages 2223–2232, 2017.