Testing Generated Distributions in GANs to Penalize Mode Collapse

Yanxiang Gong[1] **Zhiwei Xie**[1] **Mei Xie**[1,*] **Xin Ma**[2,*]
[1] School of Information and Communiaction Engineering, University of Electronic Science and Technology of China
[2] School of Life Science and Technology, University of Electronic Science and Technology of China
* Corresponding authors: {mxie, xm24}@uestc.edu.cn

Abstract

Mode collapse remains the primary unresolved challenge within generative adversarial networks (GANs). In this work, we introduce an innovative approach that supplements the discriminator by additionally enforcing the similarity between the generated and real distributions. We implement a one-sample test on the generated samples and employ the resulting test statistic to penalize deviations from the real distribution. Our method encompasses a practical strategy to estimate distributions, compute the test statistic via a differentiable function, and seamlessly incorporate test outcomes into the training objective. Crucially, our approach preserves the convergence and theoretical integrity of GANs, as the introduced constraint represents a requisite condition for optimizing the generator training objective. Notably, our method circumvents reliance on regularization or network modules, enhancing compatibility and facilitating its practical application. Empirical evaluations on diverse public datasets validate the efficacy of our proposed approach.

1 Introduction

Over the past years, deep generative models have achieved remarkable success, particularly excelling in image generation and translation tasks. In order to effectively capture the underlying data distribution, researchers have introduced various approaches, including likelihood-based models (Dinh et al. 2014, Kingma & Welling 2013), implicit models (Goodfellow et al. 2014), and score-based models (Ho et al. 2020). Among these, generative adversarial networks (GANs) (Goodfellow et al. 2014) stand out as a versatile family of frameworks that find extensive application in diverse domains, rendering related research highly significant.

While GANs possess various limitations, it is the notorious mode collapse issue that significantly hampers their application and progress. The captured distribution, denoted as p_g, often encompasses just a solitary or a few major modes of the real distribution, p_{data}, inadvertently disregarding the other modes. Consequently, GAN training becomes unstable, resulting in a propensity for generating images lacking diversity. To mitigate this challenge, several techniques have been introduced, including multi-generator approaches (Ghosh et al. 2018), regularization techniques (Miyato et al. 2018), and diversity penalties (Pei et al. 2021). However, as image resolutions escalate and data distributions grow more intricate, devising strategies to counteract mode collapse becomes a demanding endeavor, often struggling to keep pace with the rapid evolution of GAN-based methodologies. Hence, persistent efforts are imperative to mitigate this issue when constructing GAN-based networks for various applications. Given the competitive performances demonstrated by alternative generative frameworks, recent exploration of novel methodologies to address mode collapse could potentially empower GAN-based frameworks to stand out in competition against other robust models, such as the emerging diffusion models (Ho et al. 2020).

In the GAN framework design, the optimal discriminator is tasked with capturing both real and generated distributions, yielding a virtual training criterion for the generator that entails a Jensen-Shannon (JS) divergence between them. However, the JS divergence will be correctly minimized by the generator only if the discriminator is optimal at each training step. There-

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s).

fore, the training objective could still tend toward its optimal value even when the generated distribution possesses low support (Arora et al. 2017). This phenomenon arises due to the discriminator's typical design as a classifier, assigning each input image a label denoting real or fake. Even if every sample in a batch is identified as real, it is not guaranteed that the entire batch mirrors the distribution, as it remains unlikely to repeatedly yield samples from one or a few limited categories. Despite anticipating the discriminator's awareness of this context, we often neglect to provide corresponding designs or guidelines. From this standpoint, prevalent strategies seek to enhance the discriminator's capabilities, enabling it to consider the data distribution when discriminating samples (Lin et al. 2018, Nguyen et al. 2017), or involve additional modules to guide the generator in capturing the distribution (Bang & Shim 2021, Gong et al. 2023, Pei et al. 2021, Tran et al. 2018). However, a pivotal challenge emerges regarding how to accurately represent the distribution itself. Distributions characterizing common data, such as images, tend to be intricate, rendering precise acquisition of cumulative distribution functions (CDFs) nearly infeasible. Consequently, we often rely on network learning to automate distribution comprehension. Yet, entrusting this responsibility to the discriminator may compromise image quality within limited parameter constraints. Alternatively, training supplementary network modules to extract distributions could potentially exacerbate the inherent instability of an already precarious training process.

Given the limitations of learning-based approaches, we explore an alternative solution that does not hinge on model learning. Our approach involves a one-sample test method applied to generated samples, penalizing deviations from the real distribution as an adjunct to the discriminator. Utilizing non-parametric estimation, we gauge the real distribution and design a differentiable test statistic to optimize the generator. This method guides the generator to faithfully capture the real distribution, effectively suppressing mode collapse. Importantly, it will not disrupt GANs' convergence or theoretical integrity, as the test's null hypothesis mandates identical distributions between real and generated data, essential for the optimal generator. Not relying on regularization or network modules enhances compatibility and ease of application. For a more comprehensive assessment of mode collapse, we introduce the Stacked Text dataset with multiple modes, yielding challenging results. Experiments on this dataset and other benchmarks affirm our method's effectiveness and competitive performance.

In summary, our contributions encompass these aspects:

- We introduce the application of statistical tests within GAN training to guide the generator in accurately capturing the real distribution, thereby effectively mitigating mode collapse. It is important to note that our proposed method seamlessly aligns with the core convergence and theoretical integrity of GANs, ensuring its ease of implementation.
- We meticulously devise a robust framework for calculating the test statistic and seamlessly integrating it into the overarching training objective. Employing a non-parametric estimation technique facilitates accurate real distribution estimation. Furthermore, the incorporation of a differentiable function streamlines the calculation of the test statistic, which is subsequently integrated into the overarching training objective.
- The validation of our method through extensive experimentation across multiple datasets underscores its effectiveness and attests to its competitive performance.

2 Related Work

The cause of mode collapse is currently inconclusive. Typically, this issue is attributed to the discriminator's inadequate ability to evaluate the diversity of data. Therefore, three common solutions were suggested to solve the issue. The first is to improve the generator's architecture, enabling it to be initially able to produce diverse images. Based on this concept, multi-generators are frequently utilized (Ghosh et al. 2018, Li et al. 2021) since they are expected to focus on different modes. However, the limitation is that they are challenging to implement as they require special network architectures. The second is to improve the discriminator design (Nguyen et al. 2017, Lin et al. 2018, Liu et al. 2021) or training (Metz et al. 2017, Bińkowski et al. 2018*a*, Mao et al. 2019, Wang et al. 2020, Yu et al. 2020), which enhances its attention on the data diversity and distributions. These techniques run the risk of compromising image quality since the discriminator will focus more on the distribution and less on quality with constrained parameters. The third is to add extra modules into GANs as a supplement to the discriminator, including regularization (Kodali et al. 2017, Miyato et al. 2018, Liu et al. 2019, 2022, Pan et al. 2022), network modules (Srivastava et al. 2017, Tran et al. 2018, Grover et al. 2019, Sattigeri et al. 2019, Bang & Shim 2021), and training objective (Eghbal-zadeh et al. 2019, Choi et al. 2020, Pei et al. 2021, Yu et al. 2022, Gong et al. 2023). The modules aim at constraining the generator and forcing it to generate samples of various modes, which is easy to apply and thus widely used.

In this work, we propose applying statistical tests to GAN training. Commonly used regularization methods may undermine the convergence and theoretical soundness of GANs (Wang et al. 2020). The convergence is unaffected by our approach since it explicitly constrains the distributions, which is necessary for obtaining the optimal GAN generator. Methods utilizing network modules may not function as expected functionality in other scenarios since they are susceptible to alterations in data and training factors. In comparison, our method only needs mathematical calculations, which are data-agnostic. As a result, our method has better compatibility in various scenes. For other methods that improve the training objective, related approaches usually consider one aspect of the distribution, such as the distances and parameters. For instance, Eghbal-zadeh et al. (2019) proposed maximizing the distance ratio between generated images to increase diversity. Choi et al. (2020) proposed to access an additional reference dataset to overcome bias by an importance weight. Pei et al. (2021) presented a diversity penalty module that enforces the generator to generate images with different features when the input latent vectors are different. Yu et al. (2022) proposed an objective to maintain a certain distance between the latent code of generated data. Gong et al. (2023) proposed to additionally constrain the mean and variance of generated data in training. Compared to these techniques, our method explicitly requires the overall distributions to be identical.

Additionally, some evaluation metrics of generative models also explored the representation of data distribution, which may overlap our method. Sajjadi et al. (2018) applied a pre-trained classifier trained on natural images to compare real and generated data at a feature level. Kynkäänniemi et al. (2019) provided an improvement, which uses k-nearest neighbors to form explicit non-parametric representation of the manifolds of data. They estimate both the real and generated data distributions because they aim at evaluation after training. In contrast to the approaches mentioned above, we employ statistical test methods in our work rather than properly estimating the generated distribution because the amount of data available in a single iteration is limited. It will be discussed in Sec. 3.1.1.

3 Method

3.1 Distribution Consistency Test

3.1.1 Problem Formulation

In GAN frameworks, the generator achieves its optimum when the real and generated distributions are the same. To achieve this objective, an adversarial loss was proposed (Goodfellow et al. 2014), which is expressed as

$$\begin{aligned} V(D,G) =& \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[log(D(\boldsymbol{x}))]+ \\ & \mathbb{E}_{\boldsymbol{z}\sim p_{z}(\boldsymbol{z})}[log(1-D(G(\boldsymbol{z})))] \end{aligned} \tag{1}$$

in which D is the discriminator, G is the generator, $\boldsymbol{x}$ is a real sample, and $\boldsymbol{z}$ is a random noise vector. Then, the training objective is

$$\min_{G}[\max_{D} V(D,G) + \lambda D_{ks}]. \tag{2}$$

To maintain the convergence, we aim to design a method whose objective is requisite for the generator's optimum. A necessary condition to ensure that two distributions are the same is that their probability density functions (PDFs) are the same. Therefore, demanding that the PDF of the real and generated distributions be the same is necessary to constrain the identity of them. However, estimating a PDF requires a large number of samples in most statistical methods. Considering the generated data distribution is dynamic with the optimization of the generator, it is inevitable to use a large training batch size in one iteration, whatever method we choose to estimate the generated distribution. Given that the development tendency of generative models is to generate high-resolution images, it will be a significant application limitation that should be avoided.

To check whether the two PDFs are the same, an alternative way is to use hypothesis testing, including parametric and non-parametric tests. Due to the fact that we cannot presumptively know the distribution type for intricate data, using parametric tests is challenging. Thus, non-parametric tests are almost our only choice. Owing to the sufficient real training data, it is possible to estimate the real distribution, and we can regard the task as a one-sample test problem. Specifically, give null hypothesis $H_0: p_g = p_{data}$ and alternative hypothesis $H_1: p_g \neq p_{data}$. Then obtain a sample batch

$$\boldsymbol{X}_g = \{X_1, X_2, ..., X_b\} \sim p_g \tag{3}$$

and check whether it is from p_{data}, where p_g is the generated distribution, p_{data} is the real distribution, and b is the batch size. If $\boldsymbol{X}_g$ is from p_{data}, accept H_0; otherwise reject H_0. Here, we mainly considers univariate data in the analysis. For multivariate data, we apply the testing to each element and average the results, without special designs. This is because it could be difficult to represent the correlations between elements, which is discussed in the Supplementary Materials.

3.1.2 Test Method

The missed modes could be minor in the distribution when mode collapse occurs. Minor modes may only slightly disturb the distribution of major modes, as shown in Fig. 1a. Then, the means and variances of real and generated distributions are still similar, and Type II errors in tests may occur. To catch this kind of small difference, we desire to build our solution using a test method that is sensitive to variations in the shape of the PDFs in order.

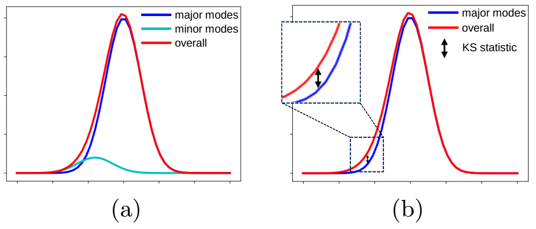


Figure 1: Illustration of data distribution with two modes. (a) shows the distributions of major modes, minor modes, and overall data. (b) shows the KS statistic between distributions of major modes and overall data. The overall data distribution is the sum of the distributions of major and minor modes.

Kolmogorov–Smirnov (KS) test is such a method, which specifies the supremum of the distances between two distributions, as shown in Fig. 1b. Thus, the method is sensitive to the difference in one point. Give the empirical cumulative distribution function of a generated sample batch

$$F_b(x) = \frac{q^g(x)}{b} \tag{4}$$

where $q^g(x)$ is the number of elements in $\boldsymbol{X}_g$ that satisfies $X \leq x$, b is the batch size, and the KS statistic is

$$D_{ks} = \sup_x |F_b(x) - P_{data}(x)| \tag{5}$$

where P_{data} is the CDF of the real distribution. When the statistic is below a threshold, the batch of samples can be regarded as sampled from p_{data} and H_0 will be accepted.

3.2 Distribution Representation

To calculate the KS statistic, $P_{data}(x)$ should be a known function. However, deriving a precise expression of $P_{data}(x)$ is difficult because the distribution type of real data is usually a complicated mixed one. To overcome this difficulty, we review the test method. In the test method, we only use the value of $P_{data}(x)$ with different x. The distribution type, parameters, and other properties are unnecessary for this task. Thus, we only need a method to estimate the values of a CDF at different points.

In non-parametric estimation methods, density estimation is an approach that can finish the task. It uses the frequency with a large number of samples to match the probability and to estimate the values of the CDF in small intervals. Commonly used methods include histogram estimation, kernel estimation, and nearest neighbor estimation. In this work, we choose histogram estimation due to its small amount of computation. More complex methods are not considered because the training time is already noticeably increased with this method, which we will analyze in the experiments.

To accomplish the task, we first split the data values into several intervals $I = \{I_i | I_i = [a_i, a_{i+1}), i \in [1, k], i \in N\}$, where k is the number of intervals. With a given real sample batch $\boldsymbol{X}_r = \{X_1, X_2, ..., X_n\}$, use q_i^r to represent the number of elements that fall into I_i. Finally, define the PDF $p_{data}(x)$ with

$$p_{data}(x) = \frac{q_i^r}{n(a_{i+1} - a_i)}, x \in I_i. \tag{6}$$

The CDF, $P_{data}(x)$, is the integral of it.

For images, the value range of the pixels is $[0, 255]$, and we uniformly split it into 51 intervals (k=51) which is effective in experiments, and ablations can be found in the Supplement Material. More intervals will increase the accuracy of estimation, but the computation will also be increased.

3.3 Differentiable Design

As $P_{data}(x)$ values are known, the KS statistic D_{ks} can be calculated. Because the estimation of $P_{data}(x)$ is not continuous, for convenience, we first calculate the number of elements in the generated sample batch $\boldsymbol{X}_g$ that falls into the interval I_i, marked as q_i^g. Then, the cumulative values $q^g(x) = \sum_i q_i^g$ can be utilized to calculate $F_b(x)$.

However, if the result is utilized to constrain the generator, all computing operations must be differentiable for effective backpropagation. When calculating q_i^g, a function similar to a band-pass filter will be applied, as shown in Fig. 2a. To make it differentiable, we use an improved radial basis function to match the filter threshold in the calculation

$$q_i^g = \sum_{j=1}^{b} exp[-0.5(\frac{X_j - \mu(i,k)}{\sigma(k)})^{10}] \tag{7}$$

where X_j is an element in $\boldsymbol{X}_g$. For data whose value range is $[0, M]$, we can define $\mu(i,k) = \frac{M(2i+1)}{2k}$ and $\sigma(k) = \frac{M}{2k}$. The improved radial basis function is illustrated in Fig. 2b. Here, we modify the exponent from 2 to 10 for a sharp cutting-off filter, making all values that are expected to be 1 larger than 0.9. Compared to a 10-step band-pass filter, this function needs less computation as only one exponentiation operation is included.

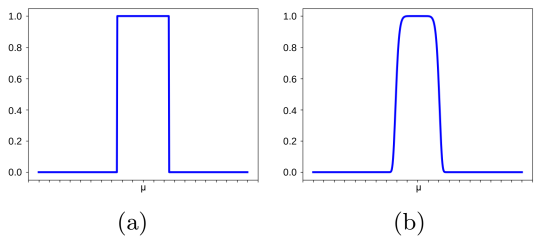


Figure 2: Illustration of (a) the non-differentiable function and (b) our differentiable function to calculate q_i^g.

3.4 Training Objective

With the differentiable design, a KS statistic D_{ks} that supports backpropagation is obtained. Instead of setting a threshold, we require the generator to minimize D_{ks}, so the confidence score will be high when the real and generated distributions are regarded as the same. Though this setting may cause more Type I errors, ablation experiments prove that it is effective.

To summarize, the training objective is

$$\min_G [\max_D V(D,G) + \lambda D_{ks}]. \tag{8}$$

Our experiments set λ to 0.1 to keep the same order of magnitude as the adversarial loss, and the ablations can be found in the Supplementary Material.

4 Experiments

We will report the critical experimental results. The implementation details, theoretical results, more ablations, and comparisons can be found in the Supplementary Material.

4.1 Stacked MNIST

To explicitly evaluate the effectiveness of suppressing mode collapse, we utilize the Stacked MNIST dataset. According to the method provided by Che et al. (2016), the images in the Stacked MNIST dataset are obtained by stacking three random single-channel images in MNIST (LeCun et al. 1998). Therefore, 1,000 modes (i.e., from 000 to 999) should be generated. We use a DCGAN architecture, provided by Lin et al. (2018), as the baseline to better reproduce mode collapse problems. We report the number of modes for which at least one sample is generated. A higher mode number denotes better results because it denotes that more modes are captured. We also report the reverse Kullback-Leibler divergence (KL) between the real and generated samples over classes. A lower KL denotes better results because it indicates that the generated distribution is more like the real one. We employ three other methods, which can be transplanted into the baseline for comparisons. The results are shown in Table 1, denoting the effectiveness of our method.

Table 1: Experiments on Stacked MNIST. Unrolled GAN is trained with 10 unrolling steps. SN denotes Spectral Norm.

Method	Modes(↑)	KL(↓)
Baseline	24	7.48
Unrolled (Metz et al. 2017)	742	0.81
SN (Miyato et al. 2018)	306	1.26
LDF (Gong et al. 2023)	970	0.23
Ours	1000	0.09

We mentioned that the discriminator might be unaware that it is a small probability event to obtain similar samples repeatedly. Thus, we additionally check the discriminator to examine its ability to deal with a sample batch from only one category, which may better denote whether mode collapse has been alleviated. We fetch the discriminator model from the last epoch where the discriminator has an advantage in the adversarial training. We get 128 real samples as a batch from each category (named monotonous samples) and separately feed them into the model. If mode collapse occurs, we assert that the discriminator will classify monotonous samples as real, ignoring that they are all from one category. If mode collapse has been alleviated, the discriminator should regard them as fake. The outputs under all categories will be averaged. Meanwhile, we also get 128 real and 128 generated samples from 128 different random categories (named diverse samples) for comparisons. Label 1 denotes real, and 0 denotes fake.

Table 2: Outputs of the discriminator from the DCGAN network on Stacked MNIST.

Method	Monotonous	Diverse	
		Real	Gen
Baseline	0.776	0.943	0.069
Unrolled, 10 steps	0.023	0.925	0.000
Spectral Norm	0.391	0.951	0.007
LDF	0.205	0.984	0.017
Ours	0.233	0.989	0.008

The results in Table 2 are in line with our conjecture. The discriminators are all close to optimal as they correctly classify diverse real and generated samples. For the baseline, the output with monotonous samples is only a bit lower but still larger than 0.5. Thus, the discriminator cannot successfully reject monotonous samples, and mode collapse could occur. For other methods, the outputs have noticeably decreased, indicating that mode collapse has been suppressed.

We must note that we have no basis to say that smaller outputs for monotonous samples represent less mode collapse when the output is already very low. For instance, Unrolled GAN provides a powerful discriminator because they propose a training strategy to improve the discriminator directly. The other methods propose additional constraints so that the discriminator is not expected to reject monotonous samples alone. Thus, it is not a direct means to compare performance, and thereby our method may still be prior according to the other metrics.

4.2 Stacked Text

The Stacked MNIST dataset allows a relatively accurate evaluation of mode collapse. However, the characters in MNIST all have a hand-writing font and no backgrounds. This kind of data is not common in the real world, and the modes are also easy to be captured. There is no targeted evaluation metric for mode collapse for natural image datasets, as there is neither a powerful pre-trained classifier to recognize generated images nor fine-grained annotations of the real data. Existing metrics, such as Inception Score (IS) (Salimans et al. 2016) and Fréchet Inception Distance score (FID) (Heusel et al. 2017), mostly take into consideration the image quality simultaneously instead of solely considering the distribution. Therefore, it might be challenging to determine whether a technique prevents mode collapse or enhances image quality.

Inspired by the construction of Stacked MNIST, we explore using text character images in natural scenes to produce the stacked data. There are three advantages. First, there is enough annotated public data as text recognition has been developed for many years. Second, recognizing regular text characters is relatively simple, and many state-of-the-art methods can achieve high recognition accuracy. Third, scene text images are more similar to general natural scene images, so they can better denote the performance of a GAN-based method in application scenes.

Though the research interests are irregular text recognition recently, we still choose to use regular text images, as our aim is to examine the ability to suppress mode collapse rather than represent complicated images. Thus, we extract the alphanumeric characters from ICDAR 2013 (Karatzas et al. 2013) dataset, which has regular text with detailed location and content annotations. There are 36 categories (numbers 0-9 and alphabets a-z, case-insensitive). Then we randomly choose three character images and stack them following the rule of Stacked MNIST. There will be 46,656 (36^3) categories, which is challenging for generation tasks.

In practice, we randomly stacked 10M images, which covers 41,823 categories, and the image number of each category is different. The non-uniform distribution of modes brings more challenges. We train a ViT classifier (Dosovitskiy et al. 2020) on the gray character images, which has achieved an accuracy of 1.0 on the training data and an accuracy of 0.9 on the testing data. The code, model weights, and dataset will be publicly available.[1] Then, we still utilize the DCGAN network used in Stack MNIST experiments. Modes and KL will be reported, as shown in Table 3.

Table 3: Experiments on Stacked Text. Unrolled GAN is trained with 10 unrolling steps. The results are averaged over 10 trials with standard error reported.

Method	Modes(↑)	KL(↓)
Baseline	18068.4±785.23	0.0047±0.0014
Unrolled	17689.4±824.09	0.0242±0.0020
SN	18169.0±701.18	0.0035±0.0009
LDF	18547.5±583.88	0.0146±0.0083
Ours	19663.7±375.45	0.0061±0.0010

We observe that the performance even becomes worse using Unrolled GAN. It is because the task is too challenging for the simple network architecture, as this method only provides a training strategy. The other methods have improved the performance, but only less than half of the modes are captured. Thus, Stacked Text can represent the mode collapse issue and brings new challenges for related methods.

4.3 Natural Datasets

Baseline We choose a milestone work StyleGAN2 (Karras, Laine, Aittala, Hellsten, Lehtinen & Aila 2020) as our baseline, while the ADA enhancement (Karras, Aittala, Hellsten, Laine, Lehtinen & Aila 2020) is used. Because this framework is still widely used as the baseline recently (Sauer et al. 2022, Zhang et al. 2022), our method will benefit many works if we can achieve a competitive performance on it. Meanwhile, we also conduct experiments with DCGAN (Radford et al. 2015) and WGAN-GP (Gulrajani et al. 2017) on small datasets to better evaluate our method.

[1] https://github.com/yxgong0/Stacked_Text

Table 4: Comparisons on CIFAR-10 dataset.

Method	Baseline	IS(↑)	FID(↓)
DCGAN (Radford et al. 2015)	-	6.471	53.64
WGAN-GP (Gulrajani et al. 2017)	-	7.350	29.84
D2GAN (Nguyen et al. 2017)	Proposed	7.150	-
SNGAN (Miyato et al. 2018)	Proposed	-	29.30
MDGAN (Eghbal-zadeh et al. 2019)	Proposed	-	36.80
UniGAN (Pan et al. 2022)	Proposed	3.790	-
PDPM (Pei et al. 2021)	WGAN-GP	7.830	29.03
MaF-GAN (Liu et al. 2021)	WGAN	-	30.85
MaEM-GAN (Liu et al. 2022)	MaF-GAN	-	29.22
Dist-GAN (Tran et al. 2018)	DCGAN	-	45.60
DRAGAN (Kodali et al. 2017)	DCGAN	6.900	52.86
MGGAN (Bang & Shim 2021)	DCGAN	6.473	53.30
Ours	DCGAN	7.032	41.83
	WGAN-GP	**7.895**	**28.11**

Datasets We choose the widely used datasets, including CIFAR-10 (Krizhevsky 2009), CelebA (Liu et al. 2015), LSUN Bedroom (Yu et al. 2015) and FFHQ (Karras et al. 2019). We synthesize images for CIFAR-10 on 32×32 resolutions, CelebA and LSUN Bedroom on 64×64, and FFHQ on 256×256 to evaluate the performance at different image sizes.

Metrics Following Karras, Aittala, Hellsten, Laine, Lehtinen & Aila (2020), we report the IS (Salimans et al. 2016), FID (Heusel et al. 2017) and kernel inception distance (KID) (Bińkowski et al. 2018*b*) of 50k generated images. These metrics simultaneously evaluate the image quality and distribution coverage. The improved Recall (Kynkäänniemi et al. 2019) is also involved in examining the coverage of the distribution. The MS-SSIM in one class (Odena et al. 2017) is also reported on some datasets to evaluate the data diversity explicitly.

CIFAR-10 Experiments As some methods proposed special architectures that are not large, we use DCGAN and WGAN-GP as the baseline for comparisons as fair as possible. The results are shown in Table 4. Some methods are also pluggable and report the performance on different baselines (Arjovsky et al. 2017, Gulrajani et al. 2017, Radford et al. 2015). Our method improves the performance of baselines, which is effective and has reached mainstream performance.

CelebA and LSUN Bedroom Experiments Considering SOTA methods have achieved competitive performance on these large datasets, we choose StyleGAN2-ADA as the baseline and compare the methods with ViTGAN (Lee et al. 2021) to observe how big the gap is. The results are shown in Table 5, and our method has achieved results very close to ViT-

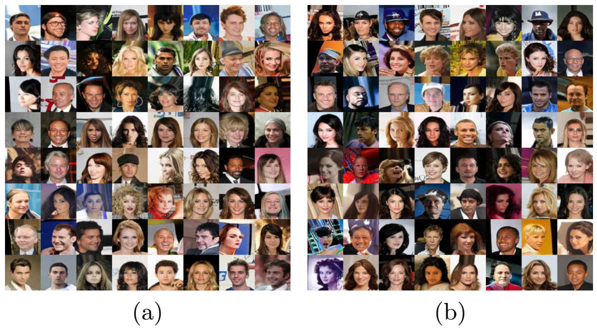


Figure 3: Generated batches on CelebA of the baseline (a) and the baseline with our method (b).

GAN, demonstrating its effectiveness. The generated image batches are also visualized, as shown in Figs. 3 and 4. We use fixed seeds from 0 to 63 to generate the batches for both models, and more modes (e.g., ethnicity and bed color) are captured with our method.

FFHQ We also try our method on 256×256 FFHQ images. The improvements on the baseline are shown in Table 5, and the visualized images (generated with fixed seeds from 0 to 15) are shown in Fig. 5. From the images, it can be observed that image quality has not been compromised. Larger images are not considered for two reasons. First, our work concentrates on suppressing mode collapse rather than struggling to achieve the SOTA performance on challenging datasets. For high-resolution images, mode collapse is not the only problem to be solved. Improving the coverage of distribution is just a drop in the bucket, so the improvements may not be obvious. Even if there is a noticeable improvement, it is hard to say whether it

Table 5: Experiments on CelebA, LSUN Bedroom, and FFHQ. Our method has a StyleGAN2-ADA baseline.

Dataset	Method	IS(↑)	FID(↓)	KIDx10^3(↓)	Recall(↑)	MS-SSIM(↓)
CelebA	StyleGAN2-ADA	3.12	2.01	0.49	0.49	0.24
	ViTGAN	**3.21**	3.74	-	-	-
	Ours	3.17	**2.00**	**0.45**	**0.53**	**0.23**
LSUN Bedroom	StyleGAN2-ADA	2.11	2.84	65.5	0.42	0.09
	ViTGAN	**2.36**	**2.65**	-	-	-
	Ours	2.29	**2.65**	**65.3**	**0.54**	**0.08**
FFHQ	StyleGAN2-ADA	5.31	3.62	1.08	0.43	0.20
	Ours	**5.35**	**3.60**	**1.00**	**0.50**	**0.17**

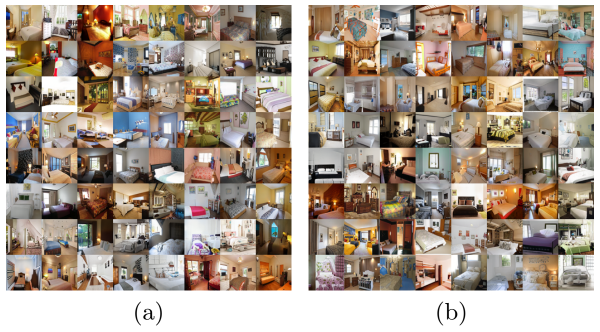


Figure 4: Generated batches on LSUN Bedroom of the baseline (a) and baseline with our method (b).

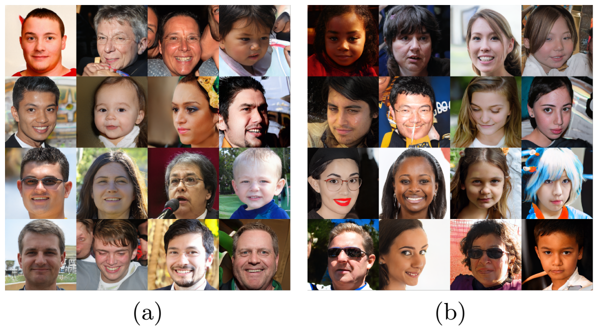


Figure 5: Generated batches on FFHQ of the baseline (a) and baseline with our method (b).

is because mode collapse has been alleviated or other reasons. Second, our method is not efficient for large images, and we will analyze this issue. However, it is worth noting that the limitation is not insurmountable, as analyzed in Sec. 4.5.

4.4 Ablations

Since our method is an improvement to GAN frameworks rather than an architecture specially designed for images, we did not extensively investigate the distribution representation of multivariate data. Instead, we focused on considering the marginal distributions of pixels, following Gong et al. (2023). Therefore, our method may not be the optimal solution for images, as it overlooks the relationships between pixels and only addresses the necessary condition of identical joint distributions. To explore whether we can achieve a better representation with features, we utilized several pre-trained feature extractors to transform images into manifolds in feature space and then applied our method. The results are presented in Table 6.

We can observe that only ViT shows a marginal improvement, but it comes with an increased computational burden. Therefore, we believe that this issue cannot be easily resolved and requires more focused design efforts. Since it is not the primary focus of this research, we provide some preliminary analysis in the Supplementary Materials.

Table 6: Feature space ablations on CIFAR-10 dataset with DCGAN.

Feature Extractor	IS(↑)	FID(↓)
Baseline (no constraint)	6.471	53.64
None (pixel space, Ours)	7.032	41.83
VGG-16 (Simonyan & Zisserman 2014)	6.999	43.01
ResNet-50 (He et al. 2016)	6.901	41.45
Inception V3 (Szegedy et al. 2016)	6.980	42.42
ViT (Dosovitskiy et al. 2020)	7.011	41.24

4.5 Efficiency and Limitation

We notice that the training time is increased with our method. Though we have applied many lightweight schemes, including the radial basis function and the simple histogram estimation, the floating-point operations (FLOPs) may still be a large number. We report results about efficiency in the Supplementary Material. When the image resolution is 256×256, the running time reaches nearly 0.5s, which takes up more than 20% of the model training time. It is because the increment of resolutions will multiply FLOPs with the increment of data elements. Thus, efficiency is the main limitation of our method. To overcome this issue, we provide two ideas for future works.

First, applying tests to features instead of images may improve efficiency. Owing to the downsampling by models, the number of data elements will decrease, resulting in less computation. However, focused design efforts are required as we need a pre-trained model that can represent the features related to modes well. Second, conducting the distribution test in part of the image is also possible. For most scenes, features related to modes are not uniformly distributed in an image. The unrelated features, e.g., the background features, could be unnecessary to be tested. Therefore, considering segmentation-based or attention-based methods to make the method focus on critical regions will improve the efficiency.

5 Discussion and Conclusion

In this study, we introduce the use of the KS test to mitigate mode collapse during GAN training. Our method effectively applies the KS test to generated samples, mitigating mode collapse while maintaining the convergence and theoretical integrity of GANs. Importantly, our approach is data-agnostic and free from prerequisites, enhancing compatibility and ease of application. However, we acknowledge efficiency limitations, particularly when dealing with high-resolution images. To address this, we propose two strategies to overcome this challenge, suggesting potential avenues for improvement. Moving forward, future research could explore more accurate estimation methods and direct divergence calculation between distributions. Learning-based methods to obtain a more accurate distribution CDF can also be attractive. It is also imperative to investigate the representation of multivariate data distributions, as GANs are typically employed for image generation.

Acknowledgements

This work was partly supported by the National Key Research and Development Program of China under Grant 2018AAA0103203, and partly supported by the Team Project of Tianfu Talent Program under Grant 2022500120654160. We would like to thank the reviewers who provided valuable comments.

References

Arjovsky, M., Chintala, S. & Bottou, L. (2017), Wasserstein generative adversarial networks, *in* 'International Conference on Machine Learning (ICML)', pp. 214–223.

Arora, S., Ge, R., Liang, Y., Ma, T. & Zhang, Y. (2017), Generalization and equilibrium in generative adversarial nets (GANs), *in* 'International Conference on Machine Learning (ICML)', pp. 224–232.

Bang, D. & Shim, H. (2021), MGGAN: Solving mode collapse using manifold-guided training, *in* 'Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)', pp. 2347–2356.

Bińkowski, M., Sutherland, D. J., Arbel, M. & Gretton, A. (2018*a*), Demystifying mmd gans, *in* 'International Conference on Learning Representations'.

Bińkowski, M., Sutherland, D. J., Arbel, M. & Gretton, A. (2018*b*), 'Demystifying MMD GANs', *arXiv:1801.01401* .

Che, T., Li, Y., Jacob, A. P., Bengio, Y. & Li, W. (2016), 'Mode regularized generative adversarial networks', *arXiv:1612.02136* .

Choi, K., Grover, A., Singh, T., Shu, R. & Ermon, S. (2020), Fair generative modeling via weak supervision, *in* 'International Conference on Machine Learning (ICML)', pp. 1887–1898.

Dinh, L., Krueger, D. & Bengio, Y. (2014), 'NICE: Non-linear inependent components estimation', *arXiv:1410.8516* .

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. et al. (2020), 'An image is worth 16x16 words: Transformers for image recognition at scale', *arXiv:2010.11929* .

Eghbal-zadeh, H., Zellinger, W. & Widmer, G. (2019), Mixture density generative adversarial networks, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 5820–5829.

Ghosh, A., Kulharia, V., Namboodiri, V. P., Torr, P. H. & Dokania, P. K. (2018), Multi-agent diverse generative adversarial networks, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 8513–8521.

Gong, Y., Xie, Z., Duan, G., Ma, Z. & Xie, M. (2023), 'Distribution fitting for combating mode collapse in generative adversarial networks', *IEEE Transactions on Neural Networks and Learning Systems* pp. 1–12.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014), 'Generative adversarial nets', *Advances in Neural Information Processing Systems (NeurIPS)* **2**, 2672–2680.

Grover, A., Song, J., Kapoor, A., Tran, K., Agarwal, A., Horvitz, E. J. & Ermon, S. (2019), 'Bias correction of learned generative models using likelihood-free importance weighting', *Advances in Neural Information Processing Systems (NeurIPS)* **32**, 11058–11070.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. & Courville, A. (2017), 'Improved training of wasserstein GANs', *Advances in Neural Information Processing Systems (NeurIPS)* **30**, 5769–5779.

He, K., Zhang, X., Ren, S. & Sun, J. (2016), Deep residual learning for image recognition, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 770–778.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B. & Hochreiter, S. (2017), 'GANs trained by a two time-scale update rule converge to a local nash equilibrium', *Advances in Neural Information Processing Systems (NeurIPS)* **30**, 6627–6638.

Ho, J., Jain, A. & Abbeel, P. (2020), 'Denoising diffusion probabilistic models', *Advances in Neural Information Processing Systems (NeurIPS)* **33**, 6840–6851.

Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., i Bigorda, L. G., Mestre, S. R., Mas, J., Mota, D. F., Almazan, J. A. & De Las Heras, L. P. (2013), ICDAR 2013 robust reading competition, *in* 'International Conference on Document Analysis and Recognition (ICDAR)', pp. 1484–1493.

Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J. & Aila, T. (2020), 'Training generative adversarial networks with limited data', *Advances in Neural Information Processing Systems (NeurIPS)* **33**, 12104–12114.

Karras, T., Laine, S. & Aila, T. (2019), A style-based generator architecture for generative adversarial networks, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 4401–4410.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J. & Aila, T. (2020), Analyzing and improving the image quality of StyleGAN, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 8110–8119.

Kingma, D. P. & Welling, M. (2013), 'Auto-encoding variational bayes', *arXiv:1312.6114* .

Kodali, N., Abernethy, J., Hays, J. & Kira, Z. (2017), 'On convergence and stability of GANs', *arXiv:1705.07215* .

Krizhevsky, A. (2009), Learning multiple layers of features from tiny images, Master's thesis, University of Toronto.

Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J. & Aila, T. (2019), 'Improved precision and recall metric for assessing generative models', *Advances in Neural Information Processing Systems (NeurIPS)* **32**, 3927–3936.

LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998), 'Gradient-based learning applied to document recognition', *Proceedings of the IEEE* **86**(11), 2278–2324.

Lee, K., Chang, H., Jiang, L., Zhang, H., Tu, Z. & Liu, C. (2021), ViTGAN: Training GANs with vision transformers, *in* 'International Conference on Learning Representations (ICLR)'.

Li, W., Fan, L., Wang, Z., Ma, C. & Cui, X. (2021), 'Tackling mode collapse in multi-generator GANs with orthogonal vectors', *Pattern Recognition* **110**, 107646.

Lin, Z., Khetan, A., Fanti, G. & Oh, S. (2018), 'PacGAN: The power of two samples in generative adversarial networks', *Advances in Neural Information Processing Systems (NeurIPS)* **31**, 1498–1507.

Liu, H., Li, B., Wu, H., Liang, H., Huang, Y., Li, Y., Ghanem, B. & Zheng, Y. (2022), 'Combating mode collapse in GANs via manifold entropy estimation', *arXiv:2208.12055* .

Liu, H., Liang, H., Hou, X., Wu, H., Liu, F. & Shen, L. (2021), 'Manifold-preserved gans', *arXiv:2109.08955* .

Liu, K., Tang, W., Zhou, F. & Qiu, G. (2019), Spectral regularization for combating mode collapse in GANs, *in* 'Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)', pp. 6382–6390.

Liu, Z., Luo, P., Wang, X. & Tang, X. (2015), Deep learning face attributes in the wild, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)', pp. 3730–3738.

Mao, Q., Lee, H.-Y., Tseng, H.-Y., Ma, S. & Yang, M.-H. (2019), Mode seeking generative adversarial networks for diverse image synthesis, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1429–1437.

Metz, L., Sohl-Dickstein, J., Poole, B. & Pfau, D. (2017), Unrolled generative adversarial networks, *in* 'International Conference on Learning Representations (ICLR)'.

Miyato, T., Kataoka, T., Koyama, M. & Yoshida, Y. (2018), Spectral normalization for generative adversarial networks, *in* 'International Conference on Learning Representations (ICLR)'.

Nguyen, T., Le, T., Vu, H. & Phung, D. (2017), 'Dual discriminator generative adversarial nets', *Advances in Neural Information Processing Systems (NeurIPS)* **30**, 2667–2677.

Odena, A., Olah, C. & Shlens, J. (2017), Conditional image synthesis with auxiliary classifier gans, *in* 'International Conference on Machine Learning (ICML)', pp. 2642–2651.

Pan, Z., Niu, L. & Zhang, L. (2022), UniGAN: Reducing mode collapse in GANs using a uniform generator, *in* 'Advances in Neural Information Processing Systems (NeurIPS)', Vol. 35, pp. 37690–37703.

Pei, S., Da Xu, R. Y., Xiang, S. & Meng, G. (2021), 'Alleviating mode collapse in GAN via diversity penalty module', *arXiv:2108.02353* .

Radford, A., Metz, L. & Chintala, S. (2015), 'Unsupervised representation learning with deep convolutional generative adversarial networks', *arXiv:1511.06434* .

Sajjadi, M. S., Bachem, O., Lucic, M., Bousquet, O. & Gelly, S. (2018), 'Assessing generative models via precision and recall', *Advances in Neural Information Processing Systems (NeurIPS)* **31**, 5234–5243.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A. & Chen, X. (2016), 'Improved techniques for training GANs', *Advances in Neural Information Processing Systems (NeurIPS)* **29**, 2234–2242.

Sattigeri, P., Hoffman, S. C., Chenthamarakshan, V. & Varshney, K. R. (2019), 'Fairness GAN: Generating datasets with fairness properties using a generative adversarial network', *IBM Journal of Research and Development* **63**(4/5), 3–1.

Sauer, A., Schwarz, K. & Geiger, A. (2022), StyleGAN-XL: Scaling StyleGAN to large diverse datasets, *in* 'ACM SIGGRAPH 2022 Conference Proceedings', pp. 1–10.

Simonyan, K. & Zisserman, A. (2014), 'Very deep convolutional networks for large-scale image recognition', *arXiv:1409.1556* .

Srivastava, A., Valkov, L., Russell, C., Gutmann, M. U. & Sutton, C. (2017), 'VEEGAN: Reducing mode collapse in GANs using implicit variational learning', *Advances in Neural Information Processing Systems (NeurIPS)* **30**, 3310–3320.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. (2016), Rethinking the inception architecture for computer vision, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 2818–2826.

Tran, N.-T., Bui, T.-A. & Cheung, N.-M. (2018), Dist-GAN: An improved gan using distance constraints, *in* 'Proceedings of the European Conference on Computer Vision (ECCV)', pp. 370–385.

Wang, D., Qin, X., Song, F. & Cheng, L. (2020), 'Stabilizing training of generative adversarial nets via langevin stein variational gradient descent', *IEEE Transactions on Neural Networks and Learning Systems* **33**(7), 2768–2780.

Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T. & Xiao, J. (2015), 'LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop', *arXiv:1506.03365* .

Yu, N., Li, K., Zhou, P., Malik, J., Davis, L. & Fritz, M. (2020), Inclusive GAN: Improving data and minority coverage in generative models, *in* 'Proceedings of the European Conference on Computer Vision (ECCV)', pp. 377–393.

Yu, S., Zhang, K., Xiao, C., Huang, J. Z., Li, M. J. & Onizuka, M. (2022), 'HSGAN: Reducing mode collapse in GANs by the latent code distance of homogeneous samples', *Computer Vision and Image Understanding* **214**, 103314.

Zhang, B., Gu, S., Zhang, B., Bao, J., Chen, D., Wen, F., Wang, Y. & Guo, B. (2022), StyleSwin: Transformer-based gan for high-resolution image generation, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 11304–11314.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes.]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes. Detailed running time can be found in the supplementary materials.]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes. They will be publicly available in the URL given in Sec. 4.2.]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Not Applicable.]

   (b) Complete proofs of all theoretical results. [Not Applicable.]
   (c) Clear explanations of any assumptions. [Not Applicable.]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes.]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes.]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes.]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes.]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes.]
   (b) The license information of the assets, if applicable. [Not Applicable.]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes. They will be publicly available in the URL given in Sec. 4.2.]
   (d) Information about consent from data providers/curators. [Not Applicable.]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable.]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable.]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable.]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable.]

Supplementary Materials - Testing the Generated Distribution in GANs to Penalize Mode Collapse

1 Implementation Details

1.1 Stacked MNIST and Stacked Text Experiments

The detailed DCGAN architectures provided by Lin et al. (2018) are described in Tables 1 and 2.

Layers	Configurations	Channels
Fully Connected	-	256-1024
Reshape	-	1024-64
Batch Normalization	-	-
ReLU	-	-
Transposed Convolution	{k3,s2,p1}	64-32
Batch Normalization	-	-
ReLU	-	-
Transposed Convolution	{k3,s2,p1}	32-16
Batch Normalization	-	-
ReLU	-	-
Transposed Convolution	{k3,s2,p1}	16-8
Batch Normalization	-	-
ReLU	-	-
Transposed Convolution	{k3,s1,p1}	8-3
Tanh	-	-

Table 1: Architecture of the generator in Stacked MNIST and Stacked Text experiments. k, s, and p represent kernel size, stride, and padding size. Channels show the input-output channels.

Layers	Configurations	Channels
Convolution	{k3, s2, p1}	3-8
Batch Normalization	-	-
Leaky ReLU	0.3	-
Convolution	{k3, s2, p1}	8-16
Batch Normalization	-	-
Leaky ReLU	0.3	-
Convolution	{k3, s2, p1}	16-32
Batch Normalization	-	-
Leaky ReLU	0.3	-
Reshape	-	32-512
Fully Connected	-	512-1
Sigmoid	-	-

Table 2: Architecture of the discriminator in Stacked MNIST and Stacked Text experiments. k, s, and p represent kernel size, stride, and padding size. Channels show the input-output channels.

In the training process, the optimizer is Adam Kingma & Ba (2014) with a learning rate 0.0002 for both the generator and the discriminator. The batch size is 128, and the network is trained for 200 epochs. The experiments are carried out on a Linux platform with a single NVIDIA TITAN Xp GPU. Note that the settings are used for all models, including models with other enhancement methods (unrolled, spectral norm, and LDF), and both datasets.

In the testing process, for Stacked MNIST, each model will generate 25,600 samples and we compute the predicted class label of each image channel by a pre-trained MNIST classifier[1]. For Stacked Text, each model will generate 204,800 images and we compute the predicted class label by a ViT classifier trained by ourselves, as mentioned in the paper.

Following Gong et al. (2023), we regard each image pixel as a data element and apply tests to each pixel. The results will be averaged. We will also analyze and evaluate the effectiveness of operating manifolds at feature space in Sec. 4.3.

In the experiments of discriminator analysis, the adjunct is not included in the discriminator. In other words, the adjunct, such as LDF and our KS statistic, will not impact the outputs of rejecting monotonous samples. Therefore, the performance of rejecting monotonous samples represents the single ability of the discriminator.

1.2 Natural Dataset Experiments

When using DCGAN Radford et al. (2015) as the baseline, the optimizer is Adam Kingma & Ba (2014) with a learning rate 0.001. The batch size is 128, and the networks are trained for 200 epochs. The experiments are conducted on a Linux platform with a single NVIDIA TITAN Xp GPU.

When using WGAN-GP Gulrajani et al. (2017) as the baseline, the optimizer is Adam Kingma & Ba (2014) with a learning rate 0.00005. The batch size is 128, and the networks are trained for 200 epochs. The experiments are conducted on a Linux platform with a single NVIDIA TITAN Xp GPU.

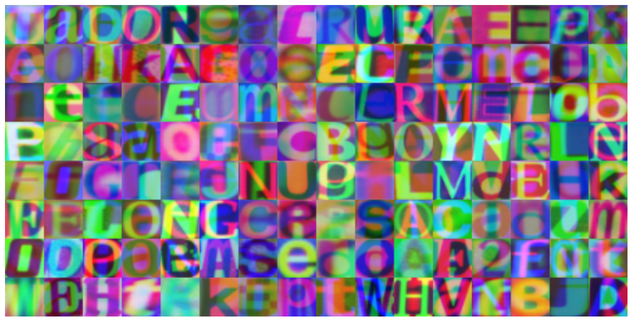
When using StyleGAN2-ADA Karras et al. (2020) as the baseline, the optimizer is Adam Kingma & Ba (2014) with a learning rate 0.001. The batch size is automatically set with the official code[2] (64 for CIFAR-10, 32 for CelebA and LSUN bedroom, and 16 for FFHQ). The networks are trained for 100,000 iterations for CIFAR-10 and 25,000 iterations for the other datasets. The experiments are conducted on a Linux platform with two NVIDIA TITAN RTX GPUs. According to the methods provided by Karras et al. (2020), the balanced consistency regularization (bCR) is used; class-conditional settings are not used; the CIFAR-specific architecture tuning is utilized for CIFAR-10.

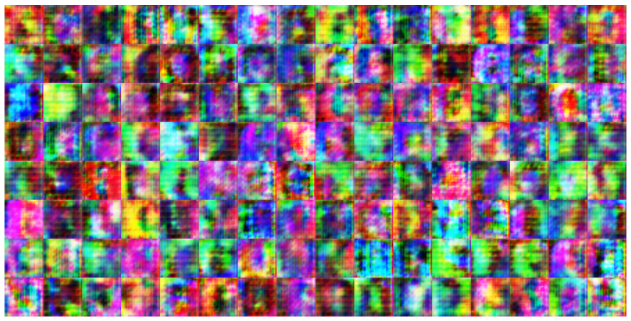
Similarly, all models regard image pixels as data elements. The methods for comparisons in the paper include D2GAN Nguyen et al. (2017), SNGAN Miyato et al. (2018), MDGAN Eghbal-zadeh et al. (2019), UniGAN Pan et al. (2022), PDPM Pei et al. (2021), MaF-GAN Liu et al. (2021), MaEM-GAN Liu, Li, Wu, Liang, Huang, Li, Ghanem & Zheng (2022), Dist-GAN Tran et al. (2018), DRAGAN Kodali et al. (2017), MGGAN Bang & Shim (2021), and ViTGAN Lee et al. (2021).

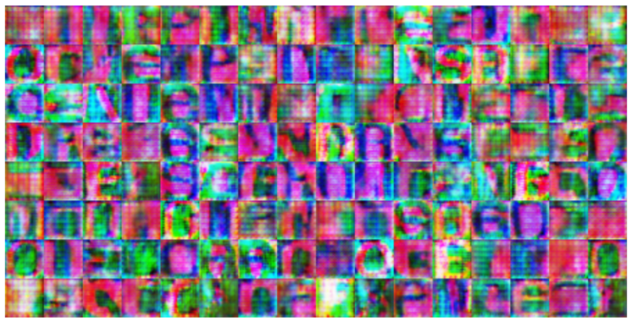
1.3 Metric Details

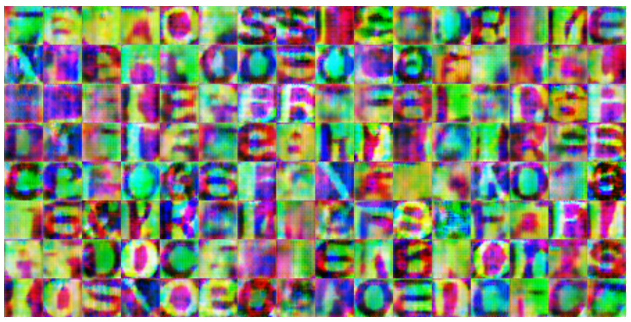
The evaluation metrics are calculated with the same approach as the one utilized by Karras et al. (2020). The IS is calculated with 50k generated images. The FID, KID, and Recall are calculated with 50k generated images against the full real dataset. The MS-SSIM is calculated with 100 randomly chosen pairs of images, following Odena et al. (2017). The metrics are implemented with public code for MS-SSIM[3] and public code for the others[4]. Higher IS, lower FID and KID represent better image quality and better distribution coverage. Higher Recall represents better distribution coverage. Lower MS-SSIM represents better diversity because it denotes that the images are less similar.

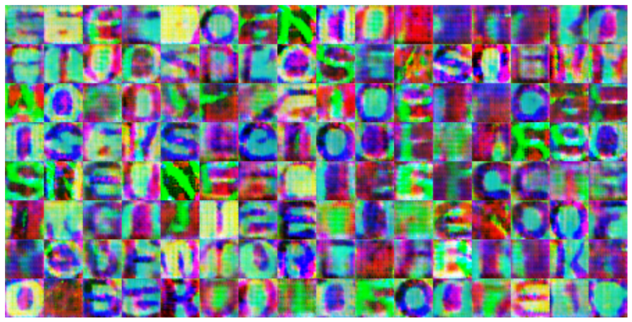
[1]https://github.com/fjxmlzn/PacGAN
[2]https://github.com/NVlabs/stylegan2-ada-pytorch
[3]https://github.com/VainF/pytorch-msssim
[4]https://github.com/NVlabs/stylegan2-ada-pytorch


(a) Real samples


(b) Baseline samples


(c) Unrolled samples (10 steps)


(d) Spectral Norm samples


(e) LDF samples

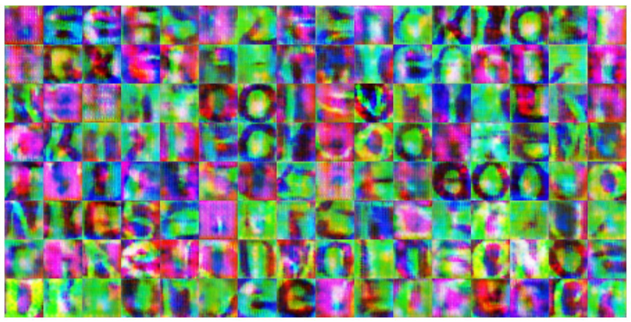

(f) Our samples

Figure 1: Real and generated images for Stacked Text.

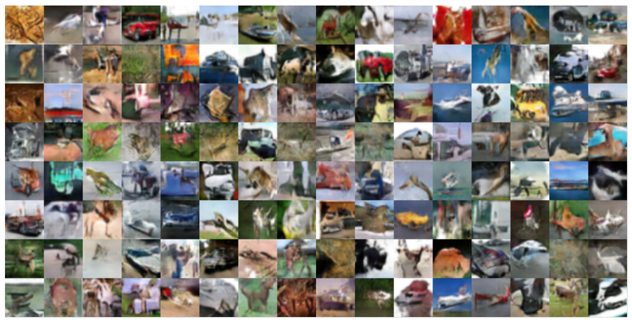
2 Visualized Samples

The visualization of Stacked Text samples is shown in Fig. 1. Each model randomly generates 128 samples. The visualization of CIFAR-10 samples is shown in Fig. 2. Because there are only 10 categories in CIFAR-10, the diversity cannot be intuitively observed. However, it can be observed that the image quality is not sacrificed.

The loss curves of the DCGAN training on CIFAR-10 are shown in Fig. 3. Although our method is not designed for enhancing stability, the training is still stable. The reason could be that our method imposes a necessary condition on the optimal generator for effective GAN training. Deep learning models, with sufficient capacity, can typically optimize both our penalty and the GAN objective simultaneously, indicating most models can be guided correctly. If instability arises, suggesting potential mode collapse, additional stability-enhancing techniques may be needed.

3 Theoretical Results

We will give some theoretical results for univariate data to further support the claim that our method keeps the convergence and theoretical soundness of GANs. In the tests, the null hypothesis is $H_0 : p_g = p_{data}$. The


(a) Baseline (DCGAN) samples

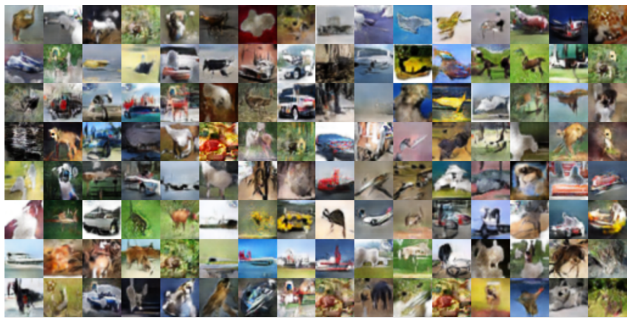
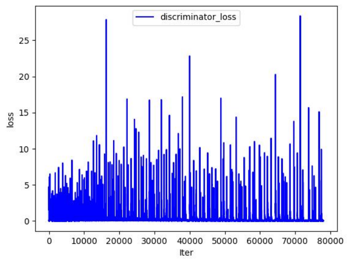
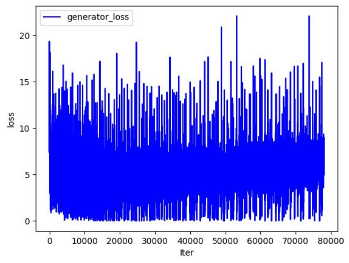

(b) Our samples

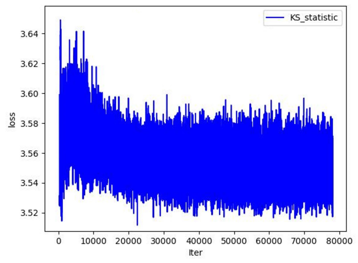
Figure 2: Generated images for CIFAR-10.



(a) Losses of the discriminator.



(b) Losses of the generator.



(c) The KS statistics in training.

Figure 3: Loss curves of DCGAN training on CIFAR-10.

training objective of our method is

$$\min_G[\max_D V(D,G) + \lambda D_{ks}]. \tag{1}$$

For simplicity, use $C(G) = \max_D V(D,G)$ to represent the original virtual training criterion for the generator G, and use $C^*(G) = C(G) + \lambda D_{ks}$ to represent our virtual training criterion for G. Goodfellow et al. (2014) have proved that if and only if $p_g = p_{data}$, $C(G)$ can reach the minimum. Then we will prove that $C^*(G)$ can reach the minimum with the same condition.

Lemma 1. *For a function $f(x) = f_1(x) + f_2(x)$, $f(x)$ must be bounded below if $f_1(x)$ and $f_2(x)$ are both bounded below.*

Proof. Assume $f_1(x) \geq M$ and $f_2(x) \geq N$, then there must be $f(x) \geq M + N$. Therefore, $f(x)$ must be bounded below. □

Lemma 2. *For a function $f(x) = f_1(x) + f_2(x)$ where $f_1(x)$ and $f_2(x)$ are both bounded below, if $f_1(x)$ and $f_2(x)$ can reach the minimum simultaneously, $f(x)$ will reach the minimum if and only if $f_1(x)$ and $f_2(x)$ reach the minimum simultaneously.*

Proof. Assume $f_1(x) \geq M$ and $f_2(x) \geq N$, then there must be $f(x) \geq M + N$. If there exists x which make $f_1(x) = M$ and $f_2(x) = N$, we will have $f(x) = M + N$, which is the minimum. Therefore, $f(x)$ will reach the minimum if $f_1(x)$ and $f_2(x)$ reach the minimum simultaneously.

Assume there exists x which makes $f(x) = M + N$ when $f_1(x)$ and $f_2(x)$ do not reach the minimum simultaneously. Then there must be $f_1(x) > M$ or $f_2(x) > N$. If $f_1(x) > M$, we have

$$\begin{aligned} f_2(x) &= M + N - f_1(x) \\ &= N + [M - f_1(x)] \\ &< N \end{aligned} \tag{2}$$

which is contrary with $f_2(x) \geq N$. If $f_2(x) > N$, similarly we have $f_1(x) < M$ which is contrary with $f_1(x) \geq M$. Therefore, $f(x)$ will reach the minimum only if $f_1(x)$ and $f_2(x)$ reach the minimum simultaneously. $\square$

Theorem 1. *If and only if $p_g = p_{data}$, $C^*(G)$ can reach the minimum.*

Proof. $C(G)$ is bounded below according to the results provided by Goodfellow et al. (2014). The KS statistic D_{ks} is non-negative according to its definition. Then $C(G)$ and D_{ks} are both bounded below. According to Lemma 1, $C^*(G)$ is also bounded below.

When $p_g = p_{data}$, $C(G)$ reaches the minimum according to the results provided by Goodfellow et al. (2014), and it is possible that H_0 is accepted. Therefore, $C(G)$ and D_{ks} can reach the minimum simultaneously, and $C^*(G)$ can reach the minimum according to Lemma 2.

According to Lemma 2, $C^*(G)$ will reach the minimum only if $C(G)$ and D_{ks} reach the minimum simultaneously. According to the results provided by Goodfellow et al. (2014), $C(G)$ will reach the minimum only if $p_g = p_{data}$. Therefore, $C^*(G)$ will reach the minimum only if $p_g = p_{data}$. $\square$

Definition 1. *In hypothesis testing, Type I error is to reject H_0 while it is true, and Type II error is to accept H_0 while it is false.*

Theorem 2. *When Type I error occurs, $C^*(G)$ cannot reach the minimum even if $p_g = p_{data}$.*

Proof. When Type I error occurs, H_0 must be rejected. Therefore, D_{ks} must have a non-zero value in our design. Therefore, even if $p_g = p_{data}$, which causes $C(G)$ reaches the minimum, $C^*(G)$ still cannot reach the minimum. $\square$

Theorem 3. *Even if Type II error occurs, $C^*(G)$ cannot reach the minimum.*

Proof. When Type II error occurs, H_0 must be accepted. Therefore, D_{ks} must reach zero. However, $C(G)$ cannot reach the minimum if $p_g \neq p_{data}$ according to the results provided by Goodfellow et al. (2014). Therefore, $C^*(G)$ cannot reach the minimum. $\square$

The results represent that only Type I errors will impact the convergence of GANs, and Type II errors will only cause the method to not work. When Type I error occurs, the generator should learn to not only make $p_g = p_{data}$ but also overcome Type I error. However, the generator should still keep $p_g = p_{data}$ when overcoming Type I errors. Therefore, the optimum is not changed, so the convergence and technical soundness of GANs is not broken.

To overcome Type I error, the generator will tend to generate samples that totally obey the real distribution. Thus, over-fitting risks will increase. However, the probability of Type I error (the level of significance α) is very small. Meanwhile, repeatedly testing in the training iterations will further reduce the probability of the error. We also do not observe negative effects in experiments at Sec. 4.1. Considering that over-fitting is not a noticeable problem in GANs, our method is still effective. However, aggravating over-fitting is a possible potential limitation of our method, which should be noted.

4 Ablations

4.1 Threshold Analysis

We have reported ablation results of the threshold for D_{ks}, as shown in Table 3.

The performance becomes worse with the increase of T. In theory, a larger T will decrease the probability of Type I errors, while D_{ks} is easier to reach 0. Thus, a smaller T will require the generator to learn to overcome Type I errors, which increase over-fitting risks according to our analysis. Meanwhile, a larger T denotes the relaxation of the constraint. A larger T will cause the method to not work more frequently. According to the results, the mode collapse problem could be more severe than the over-fitting problem in GAN training. As a result, we set T to 0 for the best performance.

Threshold (T)	IS(↑)	FID(↓)
Baseline (no constraint)	6.471	53.64
0 (Ours)	7.032	41.83
0.05	6.894	41.90
0.23	6.865	45.11
0.5	6.490	51.98

Table 3: Threshold ablations on CIFAR-10 dataset with DCGAN.

4.2 Number of Intervals

For images, the value range of one pixel is $[0, 255]$, and we uniformly split it into 51 intervals (k=51). We try some typical numbers to evaluate the effectiveness, as shown in Table 4.

#Intervals (k)	Interval Len	IS(↑)	FID(↓)
Baseline (no constraint)	-	6.471	53.64
5	51	6.554	52.32
17	15	6.799	43.20
51 (Ours)	5	7.032	41.83
85	3	7.100	41.89
255	1	7.098	41.30

Table 4: Interval ablations on CIFAR-10 dataset with DCGAN.

It can be seen that when using more than 51 intervals, the improvements will not be obvious. Meanwhile, the computation will be increased, so we consider 51 to be a proper number for images in the experiments.

4.3 Feature Space for Images

As our method is an improvement for GAN frameworks instead of an architecture specially designed for images, we do not thoroughly research the distribution representation of multivariate data and simply follows Gong et al. (2023). However, we will give some preliminary discussion and results for future works.

In experiments, we regard original image pixels as data elements to apply tests. For multivariate data, it is almost impossible to estimate the joint distribution due to the complexity of relations between dimensions. Admittedly, solely considering each pixel can enable us to obtain a necessary condition of $p_g = p_{data}$ since having the same marginal distribution is a necessary condition of having the same joint distribution. However, inspired by perceptual loss Johnson et al. (2016), it seems that there is a simple method to take into consideration relations between pixels, which is to use pre-trained models to transform images into feature maps at first.

However, we have some concerns about using feature maps for tests. First, commonly used pre-trained models are not designed for distribution estimation. Use $\boldsymbol{I}$ to denote the values in pixel space and $\boldsymbol{M}$ to denote the values in feature space, and the process of feature extraction can be expressed as $\boldsymbol{M} = F(\boldsymbol{I})$ where F is a feature extraction model, where $\boldsymbol{M}$ is still multivariate. Even if we map the overall image into one value M, we can only estimate $P_M = P_{F(\boldsymbol{I})}$ instead of the joint distribution $P_{\boldsymbol{I}}$, where P_ξ denotes the CDF of ξ. Thus, a special design must be applied, requiring special training and reducing compatibility. Second, the definition of mode in mode collapse is not explicit. For instance, if we train a generative model on text character images, mode collapse could not only denote that not all character classes are captured but also denote that some attributes, such as fonts, colors, and shadows, are not totally captured. The widely used feature extractor, like VGG-16 Simonyan & Zisserman (2014), are designed for general classification. Thus, these fine-grained features might be not extracted well.

We involve some commonly used pre-trained models to conduct ablations about pixel space and feature space, and the results are shown in Table 6. The models are pre-trained on ImageNet Deng et al. (2009). Our

Method	Baseline	Resolution	IS(↑)	FID(↓)
DCGAN	-	64x64	2.11	24.23
WGAN-GP	-	64x64	2.78	33.48
MaF-GAN (Liu et al. 2021)	WGAN (Arjovsky et al. 2017)	256x256	-	12.43
MGO-GAN (Li et al. 2021)	Proposed	128x128	-	189.19
Dist-GAN (Tran et al. 2018)	DCGAN (Radford et al. 2015)	32x32	-	23.70
DRAGAN (Kodali et al. 2017)	INFO GAN (Chen et al. 2016)	32x32	-	42.30
PDPM (Pei et al. 2021)	WGAN-GP (Gulrajani et al. 2017)	64x64	2.94	24.86
HSGAN (Yu et al. 2022)	Proposed	64x64	2.73	17.49
Ours	DCGAN (Radford et al. 2015)	64x64	2.92	21.19
	WGAN-GP (Gulrajani et al. 2017)	64x64	3.15	22.98

Table 5: Comparisons for CelebA dataset.

method will be applied to the output feature maps from these models. We can observe that all models achieve improvements compared to the baseline. It denotes that more modes are captured, so the performance becomes better. For different models, even the best ViT only achieves a similar result as ours, while the computation burden becomes very heavy. Thus, using feature maps may require further design to achieve a competitive performance. Considering it is not the main research interest in this work, we uphold to operate images at pixel space.

Feature Extractor	IS(↑)	FID(↓)
Baseline (no constraint)	6.471	53.64
None (pixel space, Ours)	7.032	41.83
VGG-16 (Simonyan & Zisserman 2014)	6.999	43.01
ResNet-18 (He et al. 2016)	6.754	42.98
ResNet-50 (He et al. 2016)	6.901	41.45
Inception V3 (Szegedy et al. 2016)	6.980	42.42
ViT (Dosovitskiy et al. 2020)	7.011	41.24

Table 6: Space ablations on CIFAR-10 dataset with DCGAN.

4.4 Radial Basis Function

We mentioned that in the radial basis function (RBF), we modify the exponent from 2 to 10 for a sharp cutting-off filter, making all values that are expected to be 1 larger than 0.9. If we use a smaller exponent, the fitting will be more inaccurate, which might impact the performance. To evaluate the impacts, we try some typical values, as shown in Table 7. It can be observed that 10 is a proper number since larger numbers only achieves marginal improvements.

Exponent in RBF	IS(↑)	FID(↓)
Baseline (no constraint)	6.471	53.64
2 (Initial)	6.302	55.55
4	6.477	52.15
6	6.699	45.13
8	6.808	43.00
10 (Ours)	7.032	41.83
12	7.032	41.80
14	7.044	41.85

Table 7: Exponent ablations in the RBF on CIFAR-10 dataset with DCGAN.

Abc23v!

4.5 Training Objective

The weight of the KS statistic λ is set to 0.1 to keep the same order of magnitude as the adversarial loss. To evaluate the impacts, we try different values that make the statistics to achieve different orders of magnitude, as shown in Table 8. When λ is 0.01, the performance decreases because the gradients becomes small, which could not sufficiently help the discriminator. For the other values, there are only fluctuations, indicating the effectiveness of our method.

λ	IS(↑)	FID(↓)
Baseline (no constraint)	6.471	53.64
0.01	6.987	43.11
0.1 (Ours)	7.032	41.83
1	7.004	41.81
10	7.017	41.98

Table 8: Ablations of λ on CIFAR-10 dataset with DCGAN.

4.6 Efficiency

The running time and floating-point operations (FLOPs) of our method are reported in Table 9. The baseline is StyleGAN2-ADA Karras et al. (2020). For a low image resolution, the efficiency is tolerable as the initial training time is also not a large number. For a high resolution, the running time reaches nearly 0.5s, while training StyleGAN2-ADA with high resolution is also very slow. Thus, the efficiency is a limitation of our method, especially for high-resolution image generation.

Resolution	FLOPs (M)	Running time (ms)	%
32×32	2.78	38.82	14.7
64×64	11.13	54.60	15.4
256×256	178.13	448.83	20.2

Table 9: Efficiency of our method on different resolutions. % denotes the running time of our method as a percentage of model training time.

Additionally, we did not consider the ablations of removing the discriminator for two reasons. Firstly, our method, designed for GANs, relies on the discriminator. Removing the discriminator would alter the framework, affecting our method's functionality. Secondly, the imprecise nature of histogram estimation means D_{ks} can only assist the discriminator and may not accurately represent distributions, potentially leading to unfavorable outcomes.

5 Comparisons

5.1 CelebA Results with DCGAN

Although there are many works for suppressing mode collapse using CelebA Liu et al. (2015) for experiments, fair comparisons on this dataset are hard to be reported as the setups are different. Here we list the methods and their setups for more comparison results. **Note that the comparisons may be unfair.** We use DCGAN and WGAN-GP as the baselines, whose training setups are the same as the ones used for CIFAR-10, to evaluate our method on the CelebA dataset. The results are shown in Table 5.

5.2 CelebA and FFHQ Comparisons to SOTA methods

Because we claim that exploring methods for combating mode collapse could potentially empower GAN-based frameworks to stand out in competition against other robust models, we compare our results (including the baseline StyleGAN2-ADA Karras et al. (2020)) with more SOTA generative models who reported their results

Method	Venue	Type	FID(↓)
StyleGAN2-ADA Karras et al. (2020)	NeurIPS'20	GAN	2.01
NCP-VAE Aneja et al. (2021)	ICLR'21	VAE	5.25
ViTGAN Lee et al. (2021)	ICLR'22	GAN	3.74
F-PNDM Liu, Ren, Lin & Zhao (2022)	ICLR'22	Diff	2.71
DDIM Song et al. (2021) w/ Benny & Wolf (2022)	CVPR'22	Diff	4.07
ES-DDPM Lyu et al. (2022)	Arxiv'22	Diff	2.55
Ours	-	GAN	**2.00**

Table 10: Comparisons for CelebA dataset. Our method has a StyleGAN2-ADA baseline. Diff represents diffusion models.

Method	Venue	Type	FID(↓)
StyleGAN2-ADA Karras et al. (2020)	NeurIPS'20	GAN	3.62
VDVAE Child (2021)	ICLR'21	VAE	33.5
VQGAN Esser et al. (2021)	CVPR'21	GAN	9.60
LDM-4 Rombach et al. (2022)	CVPR'22	Diff	4.98
P2 Choi et al. (2022)	CVPR'22	Diff	6.92
Ours	-	GAN	**3.60**

Table 11: Comparisons for FFHQ dataset. Our method has a StyleGAN2-ADA baseline. Diff represents diffusion models.

on CelebA with resolution 64×64, including some diffusion models Liu, Ren, Lin & Zhao (2022), Lyu et al. (2022), Benny & Wolf (2022) and a VAE model Aneja et al. (2021), as shown in Table 10, and who reported their results on FFHQ Karras et al. (2019) with resolution 256×256, including some diffusion models Rombach et al. (2022), Choi et al. (2022), a GAN model Esser et al. (2021) and a VAE model Child (2021), as shown in Table 11.

References

Aneja, J., Schwing, A., Kautz, J. & Vahdat, A. (2021), NCP-VAE: Variational autoencoders with noise contrastive priors, *in* 'International Conference on Learning Representations (ICLR)'.

Arjovsky, M., Chintala, S. & Bottou, L. (2017), Wasserstein generative adversarial networks, *in* 'International Conference on Machine Learning (ICML)', pp. 214–223.

Bang, D. & Shim, H. (2021), MGGAN: Solving mode collapse using manifold-guided training, *in* 'Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)', pp. 2347–2356.

Benny, Y. & Wolf, L. (2022), Dynamic dual-output diffusion models, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 11482–11491.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I. & Abbeel, P. (2016), 'InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets', *Advances in Neural Information Processing Systems (NeurIPS)* **29**, 2180–2188.

Child, R. (2021), Very deep vaes generalize autoregressive models and can outperform them on images, *in* 'International Conference on Learning Representations (ICLR)'.

Choi, J., Lee, J., Shin, C., Kim, S., Kim, H. & Yoon, S. (2022), Perception prioritized training of diffusion models, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 11472–11481.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. & Fei-Fei, L. (2009), ImageNet: A large-scale hierarchical image database, *in* '2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 248–255.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. et al. (2020), 'An image is worth 16x16 words: Transformers for image recognition at scale', *arXiv:2010.11929* .

Eghbal-zadeh, H., Zellinger, W. & Widmer, G. (2019), Mixture density generative adversarial networks, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 5820–5829.

Esser, P., Rombach, R. & Ommer, B. (2021), Taming transformers for high-resolution image synthesis, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 12873–12883.

Gong, Y., Xie, Z., Duan, G., Ma, Z. & Xie, M. (2023), 'Distribution fitting for combating mode collapse in generative adversarial networks', *IEEE Transactions on Neural Networks and Learning Systems* pp. 1–12.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014), 'Generative adversarial nets', *Advances in Neural Information Processing Systems (NeurIPS)* **2**, 2672–2680.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. & Courville, A. (2017), 'Improved training of wasserstein GANs', *Advances in Neural Information Processing Systems (NeurIPS)* **30**, 5769–5779.

He, K., Zhang, X., Ren, S. & Sun, J. (2016), Deep residual learning for image recognition, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 770–778.

Johnson, J., Alahi, A. & Fei-Fei, L. (2016), Perceptual losses for real-time style transfer and super-resolution, *in* 'European Conference on Computer Vision (ECCV)', pp. 694–711.

Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J. & Aila, T. (2020), 'Training generative adversarial networks with limited data', *Advances in Neural Information Processing Systems (NeurIPS)* **33**, 12104–12114.

Karras, T., Laine, S. & Aila, T. (2019), A style-based generator architecture for generative adversarial networks, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 4401–4410.

Kingma, D. P. & Ba, J. (2014), 'Adam: A method for stochastic optimization', *arXiv:1412.6980* .

Kodali, N., Abernethy, J., Hays, J. & Kira, Z. (2017), 'On convergence and stability of GANs', *arXiv:1705.07215* .

Lee, K., Chang, H., Jiang, L., Zhang, H., Tu, Z. & Liu, C. (2021), ViTGAN: Training GANs with vision transformers, *in* 'International Conference on Learning Representations (ICLR)'.

Li, W., Fan, L., Wang, Z., Ma, C. & Cui, X. (2021), 'Tackling mode collapse in multi-generator GANs with orthogonal vectors', *Pattern Recognition* **110**, 107646.

Lin, Z., Khetan, A., Fanti, G. & Oh, S. (2018), 'PacGAN: The power of two samples in generative adversarial networks', *Advances in Neural Information Processing Systems (NeurIPS)* **31**, 1498–1507.

Liu, H., Li, B., Wu, H., Liang, H., Huang, Y., Li, Y., Ghanem, B. & Zheng, Y. (2022), 'Combating mode collapse in GANs via manifold entropy estimation', *arXiv:2208.12055* .

Liu, H., Liang, H., Hou, X., Wu, H., Liu, F. & Shen, L. (2021), 'Manifold-preserved gans', *arXiv:2109.08955* .

Liu, L., Ren, Y., Lin, Z. & Zhao, Z. (2022), Pseudo numerical methods for diffusion models on manifolds, *in* 'International Conference on Learning Representations (ICLR)'.

Liu, Z., Luo, P., Wang, X. & Tang, X. (2015), Deep learning face attributes in the wild, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)', pp. 3730–3738.

Lyu, Z., Xu, X., Yang, C., Lin, D. & Dai, B. (2022), 'Accelerating diffusion models via early stop of the diffusion process', *arXiv:2205.12524* .

Miyato, T., Kataoka, T., Koyama, M. & Yoshida, Y. (2018), Spectral normalization for generative adversarial networks, *in* 'International Conference on Learning Representations (ICLR)'.

Nguyen, T., Le, T., Vu, H. & Phung, D. (2017), 'Dual discriminator generative adversarial nets', *Advances in Neural Information Processing Systems (NeurIPS)* **30**, 2667–2677.

Odena, A., Olah, C. & Shlens, J. (2017), Conditional image synthesis with auxiliary classifier gans, *in* 'International Conference on Machine Learning (ICML)', pp. 2642–2651.

Pan, Z., Niu, L. & Zhang, L. (2022), UniGAN: Reducing mode collapse in GANs using a uniform generator, *in* 'Advances in Neural Information Processing Systems (NeurIPS)', Vol. 35, pp. 37690–37703.

Pei, S., Da Xu, R. Y., Xiang, S. & Meng, G. (2021), 'Alleviating mode collapse in GAN via diversity penalty module', *arXiv:2108.02353* .

Radford, A., Metz, L. & Chintala, S. (2015), 'Unsupervised representation learning with deep convolutional generative adversarial networks', *arXiv:1511.06434* .

Rombach, R., Blattmann, A., Lorenz, D., Esser, P. & Ommer, B. (2022), High-resolution image synthesis with latent diffusion models, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 10684–10695.

Simonyan, K. & Zisserman, A. (2014), 'Very deep convolutional networks for large-scale image recognition', *arXiv:1409.1556* .

Song, J., Meng, C. & Ermon, S. (2021), Denoising diffusion implicit models, *in* 'International Conference on Learning Representations (ICLR)'.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. (2016), Rethinking the inception architecture for computer vision, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 2818–2826.

Tran, N.-T., Bui, T.-A. & Cheung, N.-M. (2018), Dist-GAN: An improved gan using distance constraints, *in* 'Proceedings of the European Conference on Computer Vision (ECCV)', pp. 370–385.

Yu, S., Zhang, K., Xiao, C., Huang, J. Z., Li, M. J. & Onizuka, M. (2022), 'HSGAN: Reducing mode collapse in GANs by the latent code distance of homogeneous samples', *Computer Vision and Image Understanding* **214**, 103314.