MCGAN: Enhancing GAN Training with Regression-Based Generator Loss

Baoren Xiao[1], **Hao Ni**[1], **Weixin Yang**[2*]

[1]University College London
[2]University of Oxford
baoren.xiao.18@ucl.ac.uk, h.ni@ucl.ac.uk, wxy1290g@gmail.com

Abstract

Generative adversarial networks (GANs) have emerged as a powerful tool for generating high-fidelity data. However, the main bottleneck of existing approaches is the lack of supervision on the generator training, which often results in undamped oscillation and unsatisfactory performance. To address this issue, we propose an algorithm called Monte Carlo GAN (MCGAN). This approach, utilizing an innovative generative loss function, termed the regression loss, reformulates the generator training as a regression task and enables the generator training by minimizing the mean squared error between the discriminator's output of real data and the expected discriminator of fake data. We demonstrate the desirable analytic properties of the regression loss, including discriminability and optimality, and show that our method requires a weaker condition on the discriminator for effective generator training. These properties justify the strength of this approach to improve the training stability while retaining the optimality of GAN by leveraging strong supervision of the regression loss. Extensive experiments on diverse datasets, including image data (CIFAR-10/100, FFHQ256, ImageNet, and LSUN Bedroom), time series data (VAR and stock data), and video data, are conducted to demonstrate the flexibility and effectiveness of our proposed MCGAN. Numerical results show that the proposed MCGAN is versatile in enhancing a variety of backbone GAN models and achieves consistent and significant improvement in terms of quality, accuracy, training stability, and learned latent space.

Code — https://github.com/DeepIntoStreams/MCGAN
Extended version — https://arxiv.org/abs/2405.17191

Introduction

In recent years, Generative Adversarial Network (GAN) (Goodfellow et al. 2014) has become one of the most powerful tools for realistic image synthesis. However, the instability of the GAN training and unsatisfying performance remains a challenge. To combat it, much effort has been put into developing regularization methods, see (Gulrajani et al. 2017; Mescheder, Geiger, and Nowozin 2018; Miyato et al. 2018; Kang, Shin, and Park 2022). Additionally, as (Arjovsky and Bottou 2017) pointed out, the generator usually suffers gradient vanishing and instability due to the singularity of the denominator showed in the gradient when the discriminator becomes accurate. To address this issue, some work has been done to develop better adversarial loss (Lim and Ye 2017; Mao et al. 2017; Arjovsky, Chintala, and Bottou 2017). As a variant of GAN, conditional GAN (cGAN) (Mirza and Osindero 2014) is designed to learn the conditional distribution of target variable given conditioning information. It improves the GAN performance by incorporating conditional information to both the discriminator and generator, we hence have better control over the generated samples (Zhou et al. 2021; Odena, Olah, and Shlens 2017).

Unlike these works on the regularization method and adversarial loss, our work focuses on the generative loss function to enhance the performance of GAN training. In this paper, we propose a novel generative loss, termed as the regression loss $\mathcal{L}_R$, which reformulates the generator training as the least-square optimization task. This regression loss underpins our proposed MCGAN, an enhancement of existing GAN models achieved by replacing the original generative loss with our regression loss. This approach leverages the expected discriminator D^ϕ under the fake measure induced by the generator. Benefiting from the strong supervision lying in the regression loss, our approach enables the generator to learn the target distribution with a relatively weak discriminator in a more efficient and stable manner.

The main contributions of our paper are three folds:

- We propose the MCGAN methodology for enhancing both unconditional and conditional GAN training.
- We establish the theoretical foundation of the proposed regression loss, e.g., the discriminability, optimality, and improved training stability. A simple but effective toy example of Dirac-GAN is provided to show that our proposed MCGAN successfully mitigates the non-convergence issues of conventional GANs by incorporating regression loss.
- We empirically validate the consistent improvements of MCGAN over various GANs across diverse data types (i.e., images, time series, and videos). Our approach improves quality, accuracy, training stability, and learned latent space, showing its generality and flexibility.

Related work GANs have demonstrated their capacity to simulate high-fidelity synthetic data, facilitating data shar-

*Corresponding author.

ing and augmentation. Extensive research has focused on designing GAN models for various data types, including images (Han et al. 2018), time series (Yoon, Jarrett, and Van der Schaar 2019; Xu et al. 2020; Ni et al. 2021), and videos (Gupta, Keshari, and Das 2022). Recently, Conditional GANs (cGANs) have gained significant attention for their ability to generate synthetic data by incorporating auxiliary information (Yoon, Jarrett, and Van der Schaar 2019; Liao et al. 2024; Xu et al. 2019). For the integer-valued conditioning variable, conditional GANs can be roughly divided into two groups depending on the way of incorporating the class information: *Classification-based* and *Projection-based* cGANs (Odena, Olah, and Shlens 2017; Miyato and Koyama 2018; Kang et al. 2021; Zhou et al. 2021; Mirza and Osindero 2014; Hou et al. 2022). For the case where conditioning variable is continuous, the training of conditioning GANs is more challenging. For example, conditional WGAN suffers difficulty in estimating the conditional expected discriminator of real data due to the need for recalibration per every discriminator update (Liao et al. 2024). Attempts are made to mitigate this issue, such as conditional SigWGAN (Liao et al. 2024), which is designed to tackle this issue for time series data.

Preliminaries

Generative Adversarial Networks

Generative adversarial networks (GANs) are powerful tools for learning the target distribution from real data to enable the simulation of synthetic data. To this goal, GAN plays a min-max game between two networks: *Generator* (G) and *Discriminator* (D). Let $\mathcal{X}$ denote the target space and $\mathcal{Z}$ be the latent space. Then *Generator* G^θ is defined as a parameterised function that maps latent noise $z \in \mathcal{Z}$ to the target data $x \in \mathcal{X}$, where $\theta \in \Theta$ is the model parameter of G. *Discriminator* $D^\phi : \mathcal{X} \to \mathbb{R}$ discriminates between the real data and fake data generated by the generator.

Let μ and ν_θ denote the true measure and fake measure induced by G^θ. For generality, the objective functions of GANs can be written in the following general form:

$$\max_\phi \mathcal{L}_D(\phi;\theta) = \mathbb{E}_\mu\left[f_1(D^\phi(X))\right] + \mathbb{E}_{\nu_\theta}\left[f_2(D^\phi(X))\right],$$
$$\min_\theta \mathcal{L}_G(\theta;\phi) = \mathbb{E}_{\nu_\theta}\left[h(D^\phi(X))\right], \quad (1)$$

where f_1, f_2 and h are real-valued functions. Different choices of f_1, f_2 and h lead to different GAN models.

There are extensive studies concerned with how to measure the divergence or distance between μ and ν_θ as the improved GAN loss function, which are instrumental in stabilising the training and enhancing the generation performance. Examples include *Hinge loss* (Lim and Ye 2017), *Wasserstein loss* (Arjovsky, Chintala, and Bottou 2017), *Least squares loss* (Mao et al. 2017), *Energy-based loss* (Zhao 2016) among others. Many of them satisfy Eqn. (1).

Example 1.
- *classical GAN (Goodfellow et al. 2014):* $f_1(w) = \log(w)$ *and* $f_2(w) = -h(w) = \log(1-w)$.
- *HingeGAN (Lim and Ye 2017):* $f_1(w) = f_2(-w) = -\max(0, 1-w)$, *and* $h(w) = -w$.
- *Wasserstein GAN (Arjovsky, Chintala, and Bottou 2017) :* $f_1(w) = f_2(-w) = w$, *and* $h(w) = -w + c_\mu$, *where* $c_\mu := \mathbb{E}_{X\sim\mu}[D^\phi(X)]$.

The Wasserstein distance is linked with the mean discrepancy. More specifically, let $d_\phi(\mu,\nu)$ denote the mean discrepancy between any two distributions μ and ν associated with test function D^ϕ defined as $d_\phi(\mu,\nu) = \mathbb{E}_{X\sim\mu}[D^\phi(X)] - \mathbb{E}_{X\sim\nu}[D^\phi(X)]$. In this case, $\mathcal{L}_G(\theta;\phi)$ could be interpreted as $d_\phi(\mu,\nu_\theta)$.

Conditional GANs

Conditional GAN (cGAN) is a conditional version of a generative adversarial network that can incorporate additional information, such as data labels or other types of auxiliary data into both the generator and discriminative loss (Mirza and Osindero 2014). The goal of conditional GAN is to learn the conditional distribution μ of the target data distribution $X \in \mathcal{X}$ (i.e., image) given the conditioning variable (i.e., image class label) $Y \in \mathcal{Y}$. More specifically, under the real measure μ, $X \times Y$ denote the random variable taking values in the space $\mathcal{X} \times \mathcal{Y}$. The marginal law of X and Y are denoted by P_X and P_Y, respectively.

The conditional generator $G^\theta : \mathcal{Y} \times \mathcal{Z} \to \mathcal{X}$ incorporates the additional conditioning variable to the noise input, and outputs the target variable in $\mathcal{X}$. Given the noise distribution Z, $G^\theta(y)$ induces the fake measure denoted by $\nu_\theta(y)$, which aims to approximate the conditional law of $\mu(y) := P(X|Y=y)$ under real measure μ. The task of training an optimal conditional generator is formulated as the following min-max game:

$$\mathcal{L}_D(\phi,\theta) = \mathbb{E}_Y\left[\mathbb{E}_{\mu(y)}[f_1(D^\phi(X))] + \mathbb{E}_{\nu_\theta(y)}[f_2(D^\phi(X)]\right],$$
$$\mathcal{L}_G(\theta;\phi) = \mathbb{E}_Y\left[\mathbb{E}_{\nu_\theta(y)}[h(D^\phi(X))]\right], \quad (2)$$

where f_1, f_2 and h are real value functions as before and $\mathbb{E}_Y$ denotes that the expectation is taken over y sampled from Y. Different from the unconditional case, $\mathcal{L}_D$ and $\mathcal{L}_G$ has in the outer expectation $\mathbb{E}_{y\sim P_Y}$ due to Y being a random variable.

Monte-Carlo GAN

Methodology

In this section, we propose the Monte-Carlo GAN (MCGAN) for both unconditional and conditional data generation. Without loss of generality, we describe our methodology in the setting of the conditional GAN task.[1] Consider the general conditional GAN composed with the generator loss $\mathcal{L}_G$ (Eqn. (2)) and the discrimination loss $\mathcal{L}_D$ outlined in the last subsection. To further enhance GAN, we propose the MCGAN by replacing the generative loss $\mathcal{L}_G$ with the following novel regression loss for training the generator from the perspective of the regression, denoted by $\mathcal{L}_R$,

$$\mathcal{L}_R(\theta;\phi) := \mathbb{E}_{(x,y)\sim\mu}\left[|D^\phi(x) - \mathbb{E}_{\hat{x}\sim\nu_\theta(y)}[D^\phi(\hat{x})]|^2\right], \quad (3)$$

where the expectation is taken under the joint law μ of X and Y. We optimize the generator's parameters θ by minimizing

[1]The unconditional GAN can be viewed as the conditioning variable is set to be the empty set.

the regression loss $\mathcal{L}_R(\theta;\phi)$. We keep the discriminator loss and conduct the min-max training as before. The training algorithm of MCGAN is given in the Appendix.

The name for Monte Carlo in MCGAN is due to the usage of the Monte Carlo estimator of expected discriminator output under the fake measure. This innovative loss function reframes the conventional generator training into a mean-square optimization problem by computing the l^2 loss between real and expected fake discriminator outputs.

Next, we explain the intuition behind $\mathcal{L}_G$ and its link with optimality of conditional expectation. Let us consider a slightly more general optimization problem for $\mathcal{L}_R$:

$$\min_{f\in\mathcal{C}(\mathcal{Y},\mathbb{R})} \mathbb{E}_\mu[|D^\phi(X) - f(Y)|^2], \tag{4}$$

It is well known that the conditional expectation is the optimal l^2 estimator. So the **minimizer** to Eqn (4) is given by the conditional expectation function $f^* : \mathcal{Y} \to \mathbb{R}$, defined as $f^*(y) = \mathbb{E}_\mu[D^\phi(X)|Y = y]$. This fact motivates us to consider the conditional expectation under the fake measure, $\mathbb{E}_{\nu_\theta(Y)}[D^\phi(X)]$, as the model for the mean equation f^*. It leads to our regression loss $\mathcal{L}_R$, where we replace f by $\mathbb{E}_{\nu_\theta(Y)}[D^\phi(X)]$ in Eqn. (4).

Minimising the regression loss $\mathcal{L}_G$ enforces the conditional expectation of $D^\phi(X)$ under fake measure $\nu_\theta(Y)$ to approach that under the conditional true distribution $\mu(Y) = \mathbb{P}(X|Y)$ for any given D^ϕ. Assume that $(G^\theta)_{\theta\in\Theta}$ provides a rich enough family of distributions containing the real distribution μ. Then there exists $\theta^* \in \Theta$, which is a minimizer of $\mathcal{L}_R(\theta,\phi)$ for all discriminator's parameter ϕ, satisfying that

$$\mathbb{E}_{\mu(Y)}[D^\phi(X)] = \mathbb{E}_{\nu_{\theta^*}(Y)}[D^\phi(X)]. \tag{5}$$

It implies that no matter whether the discriminator D^ϕ achieves the equilibrium of GAN training, the regression loss $\mathcal{L}_R$ is a valid loss to optimize the generator to match its expectation of D^ϕ between true and fake measure.

Moreover, our proposed regression loss can effectively mitigate the challenge of the conditional Wassaserstain GAN (c-WGAN). To compute the generative loss of c-WGAN, one needs to estimate the conditional expectation $\mathbb{E}_{\mu(Y)}[D^\phi(X)]$. However, when the conditioning variable is continuous, it becomes computationally expensive or even infeasible due to the need for recalibration with each discriminator update. In contrast, our regression loss does not need the estimator for $\mathbb{E}_{\mu(Y)}[D^\phi(X)]$.

Comparison Between $\mathcal{L}_R$ and $\mathcal{L}_G$

In this subsection, we delve into the training algorithm of the regression loss $\mathcal{L}_R$ and illustrate its advantages of enhancing the training stability in comparison with the generator loss $\mathcal{L}_G$. For ease of notation, we consider the unconditional case. To optimize the generator's parameters θ in our MCGAN, we apply gradient-descent-based algorithms and the updating rule of θ_n is given by

$$\begin{aligned}\theta_{n+1} =& \theta_n - \lambda \frac{\partial \mathcal{L}_R}{\partial \theta}|_{\theta=\theta_n} \\ =& \theta_n - 2\lambda \underbrace{\left(\mathbb{E}_\mu[D^\phi(X)] - \mathbb{E}_{\nu_{\theta_n}}[D^\phi(X)]\right)}_{d_\phi(\mu,\nu_{\theta_n})} H(\theta_n,\phi),\end{aligned} \tag{6}$$

where λ is the learning rate and

$$H(\theta,\phi) = \mathbb{E}_{z\sim P_Z}[\nabla_\theta G^\theta(z)^T \cdot \nabla_x D^\phi(G^\theta(z))]. \tag{7}$$

Note the gradient $\frac{\partial \mathcal{L}_R}{\partial \theta}$ takes into account not only $\nabla_x D^\phi(x)$ but also $d(\mu,\nu_\theta)$ - the discrepancy between the expected discriminator outputs under two measures μ and ν_θ.

In contrast, employing the generator loss $\mathcal{L}_G$, the generator parameter θ is updated by the following formula:

$$\begin{aligned}\theta_{n+1} =& \theta_n - \lambda \mathbb{E}_{z\sim P_Z}\Big[h'(D^\phi(G^{\theta_n}(z)))\nabla_\theta G^\theta(z)^T\Big|_{\theta=\theta_n} \\ & \cdot \nabla_x D^\phi(G^{\theta_n}(z))\Big].\end{aligned} \tag{8}$$

One can see that Eqn. (8) depends on the discriminator gradients $\nabla_x D^\phi(G^{\theta_n}(z))$ heavily.

MCGAN benefits from the strong supervision of $\mathcal{L}_R$, which provides more control over the gradient behaviour during the training. When θ is close to the optimal θ^*, even if D^ϕ is away from the optimal discriminator, $d_\phi(\mu,\nu_\theta)$ would be small and hence leads to stabilize the generator training. However, it may not be the case for the generator loss as shown in Eq. (8), resulting in the instability of generator training. For example, this issue is evident for the Hinge loss where $h(x) = x$ as shown in (Mescheder, Geiger, and Nowozin 2018).

Illustrative Dirac-GAN Example

To illustrate the advantages of MCGAN, we present a toy example from (Mescheder, Geiger, and Nowozin 2018), demonstrating its resolution of the training instability in Dirac-GAN. The Dirac-GAN example involves a true data distribution that is a Dirac distribution concentrated at 0. Besides, the Dirac-GAN model consists of a generator with a fake distribution $\nu_\theta(x) = \delta(x-\theta)$ with $\delta(\cdot)$ is a Dirac function and a discriminator $D^\phi(x) = \phi x$.

We consider three different loss functions for both $\mathcal{L}_D$ and $\mathcal{L}_G$: (1) *binary cross-entropy loss* (BCE), (2) *Non-saturating loss* and (3) *Hinge loss*, resulting GAN, NSGAN and HingeGAN, respectively. In this case, the unique equilibrium point of the above GAN training objectives is given by $\phi = \theta = 0$.

In this case, the update of training GAN is simplified to

$$\begin{cases}\phi_{n+1} = \phi_n + \lambda f'(-\phi_n\theta_n)\theta_n, \\ \theta_{n+1} = \theta_n - \lambda h'(\phi_n\theta_n)\phi_n.\end{cases}$$

where f is specified as $f(x) = -\log(1+\exp(x))$. By applying MCGAN to enhance GAN training, the update rules for the model parameters θ and ϕ are modified as follows:

$$\begin{cases}\phi_{n+1} & = \phi_n + \lambda f'(\phi_n\theta_n)\theta_n, \\ \theta_{n+1} & = \theta_n - \lambda 2(\phi_n\theta_n - \phi_n c)\phi_n.\end{cases}$$

Fig. 1 (a-c) demonstrates that GAN, NSGAN and Hinge GAN all fail to converge to obtain the optimal generator parameter $\theta^* = 0$. That is because the updating scheme of θ depends heavily on the ϕ. When ϕ fails to converge to zero, θ continues to update even if it has reached zero, and the non-zero θ further encourages ϕ updating away from 0, which

results in a vicious cycle and the failure of both generator and discriminator. In contrast, Fig. 1(d) of MCGAN training demonstrates that the generator parameter θ successfully converges to the optimal value 0 thanks to the regression loss in (3) to bring the training stability of the generator. A 2D Gaussian mixture example is also provided in Appendix, showing that MCGAN can help mitigate model collapse.

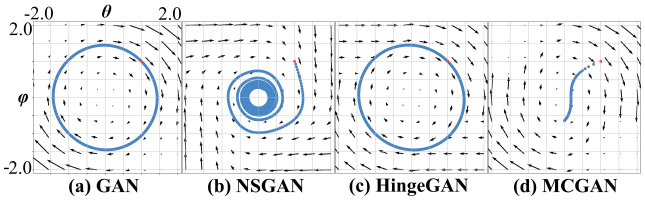


Figure 1: Dirac-GAN example

Discriminability and Optimality of MCGAN

To ensure that MCGAN training leads to the optimal generator $\nu_{\theta^*} = \mu$, one needs the sufficient discriminative power of D^ϕ. The discriminative power of D^ϕ is determined by the discriminative loss function $\mathcal{L}_D$, which is usually defined as certain divergences, such as JS divergence in GAN (Goodfellow et al. 2014). However, computing such divergence involves finding the optimal discriminator that optimizes the objective function, which might be challenging in practice. See (Liu, Bousquet, and Chaudhuri 2017) for a comprehensive description of the discriminative loss function.

Instead of needing an optimal discriminator, we introduce the weaker condition, *discriminability* of the discriminator D^ϕ, to ensure the generator's optimality for the training.

Definition 1 (Discriminability)**.** *A discriminator*

$$\mathcal{P}(\mathcal{X}) \times \mathcal{P}(\mathcal{X}) \times \mathcal{X} \to \mathbb{R}; \quad (\mu, \nu, x) \mapsto D^{\phi_{\mu,\nu}}(x),$$

where $\phi_{\cdot,\cdot} : \mathcal{P}(\mathcal{X}) \times \mathcal{P}(\mathcal{X}) \to \Phi$, is said to have discriminability if there exist two constants $a \in \{-1, 1\}$ and $c \in \mathbb{R}$ such that for any two measures $\mu, \nu \in \mathcal{P}(\mathcal{X})$, it satisfies that

$$a(D^{\phi_{\mu,\nu}}(x) - c)(p_\mu(x) - p_\nu(x)) > 0, \tag{9}$$

for all $x \in \mathcal{A}^{\mu,\nu} := \{x \in \mathcal{X} : p_\mu(x) \neq p_\nu(x)\}$. We denote the set of discriminators with discriminability as $\mathcal{D}_{Dis}$.

The discriminability of the discriminator can be interpreted as the ability to distinguish between ν and μ pointwisely over $\mathcal{A}^{\mu,\nu}$ by telling the sign (or the opposite sign) of $p_\mu(x) - p_\nu(x)$. In (9), if $a = 1$, the constant c can be regarded as a criterion in the sense that $D^{\phi_{\mu,\nu}}(x) - c$ is positive when $p_\mu(x) > p_\nu(x)$ and vice versa.

The discriminability covers a variety of optimal discriminators in GAN variants. We present in Table 1 a list of optimal discriminators of some commonly used GAN variants along with their values of a and c. The detailed description can be found in the Appendix. Although discriminability can be obtained by training the discriminator via certain $\mathcal{L}_D$, it is worth emphasizing that the discriminator does not necessarily need to reach its optimum to obtain discriminability.

Name	Discriminative loss	$D^*(x)$	a	c
Vanilla GAN	Binary cross-entropy	$\frac{p_\mu(x)}{p_\mu(x)+p_{\nu_\theta}(x)}$	1	1/2
LSGAN	Least square loss	$\frac{\alpha p_\mu(x)+\beta p_{\nu_\theta}(x)}{p_\mu(x)+p_{\nu_\theta}(x)}$	$\operatorname{sign}(\alpha-\beta)$	$\frac{\alpha+\beta}{2}$
Hinge GAN	Hinge loss	$2\mathbb{1}_{\{p_\mu(x)\geq p_{\nu_\theta}(x)\}} - 1$	1	0
Energy GAN	Energy-based loss	$m\mathbb{1}_{\{p_\mu(x)<p_{\nu_\theta}(x)\}}$	$\operatorname{sign}(-m)$	$\frac{m}{2}$
f-GAN	VLB on f-divergence	$f'\left(\frac{p_\mu(x)}{p_{\nu_\theta}(x)}\right)$	1	$f'(1)$

Table 1: List of common discriminative loss functions that satisfy strict discriminability

Assumption 1. *Let H be defined in Eqn.* (7)*. The equality $H(\theta, \phi) = \vec{0}$ holds only if (θ, ϕ) reaches the equilibrium point where $\nu_\theta = \mu$.*

Now, we establish the optimality of $\mu = \nu_\theta$ in the following theorem under the regularity condition (Assumption 1).

Theorem 1. *Assume Assumption 1 holds, and let $\phi'_{\cdot,\cdot} : \mathcal{P}(\mathcal{X}) \times \mathcal{P}(\mathcal{X}) \to \Phi$ be a parameterization map such that $D^{\phi'_{\cdot,\cdot}} : \mathcal{P}(\mathcal{X}) \times \mathcal{P}(\mathcal{X}) \times \mathcal{X} \to \mathbb{R}$ has discriminability, i.e. $D^{\phi'_{\cdot,\cdot}} \in \mathcal{D}_{Dis}$. If θ^* is a local minimizer of $\mathcal{L}_G(\theta; \phi'_{\mu,\nu_\theta}, \mu)$ defined in* (3)*, then $\nu_{\theta^*} = \mu$.*

Theorem 1 implies that MCGAN can effectively learn the data distribution μ without requiring the discriminator to reach its optimum; the discriminability is sufficient, which is again attributed to the strong supervision provided by regression loss $\mathcal{L}_R$. We defer the proof of Theorem 1 and other theoretical properties of MCGAN, e.g., improved training stability and relation to f-divergence to the Appendix.

Numerical Experiments

To validate the efficacy of the proposed MCGAN method, we conduct extensive experiments on a broad range of data, including image, time series, and video data for various generative tasks. For image generation, the conditioning variables are categorical, whereas for time series and video generation tasks, the conditioning variables are continuous. To show the flexibility of MCGAN to enhance different GAN backbones, we choose several state-of-the-art GAN models with different discriminative losses (i.e., BCE and Hinge loss) as baselines. Various test metrics and qualitative analysis are employed to give a comprehensive assessment of the quality of synthetic data generation.

The full implementation details of numerical experiments, including models, test metrics, hyperparameters, optimizer and supplementary numerical results, can be found in Appendix. Moreover, we will open-source the codes and final checkpoints upon publication for reproducibility.

Unconditional and Conditional Image Generation

Datasets We conduct conditional image generation tasks using the CIFAR-10 and CIFAR-100 datasets (Alex 2009), which are standard benchmarks with 60K 32x32 RGB images across 10 and 100 classes, respectively. To further validate our MC method on larger and higher-resolution datasets, we include: 1) the unconditional FFHQ256 dataset,

which contains 70K 256x256 human face images, 2) the conditional ImageNet64 dataset, which has 1.2 million 64x64 images across 1,000 classes, and 3) the unconditional LSUN bedroom data, which has 3 million 256x256 images.

We validate our method using two different backbones, BigGAN (Brock, Donahue, and Simonyan 2018) and StyleGAN2 (Karras et al. 2020b). The test metrics include *Inception Score* (IS), *Fréchet Inception Distance* (FID), and *Intra Fréchet Inception Distance* (IFID) together with two recognizability metrics *Weak Accuracy* (WA) and *Strong Accuracy* (SA). To alleviate the overfitting and improve the generalization, we also increase data efficiency by using the *Differentiable Augmentation* (DiffAug) (Zhao et al. 2020).

We focus on the CIFAR-10 for in-depth analysis, with a brief summary of the results on the other datasets.

Faster Training Convergence In Figure 2, we plot the learning curves in terms of FID and IS during the training. It shows that the MC method tends to have much faster convergence and ends at a considerably better level in both baselines of using Hinge loss and BCE loss.

Improved Fidelity Metrics As shown in Table 2, our MC method considerably improves all the baselines independently of the choice of discriminative loss ($\mathcal{L}_D$). Specifically, when using Hinge loss as $\mathcal{L}_D$ along with DiffAug, the MC method improves the FID from 4.43 to 3.61, comparable to the state-of-the-art FID result of (Kang, Shin, and Park 2022). Also, its IS score is significantly increased from 9.61 to 9.96, indicating better diversity of the generated samples.

In addition, applying the MC method to the cStyleGAN2 backbone results in an FID improvement of approximately 0.08. Notably, the combination of Hinge + MC + DiffAug achieves an FID of 2.16, which, to our knowledge, is the best FID achieved using StyleGAN2 as the backbone (Kang et al. 2021; Kang, Shin, and Park 2022; Tseng et al. 2021)

Loss	Hinge			BCE		
Metrics	IS ↑	FID ↓	IFID ↓	IS ↑	FID ↓	IFID ↓
BigGAN	9.27	5.31	16.20	9.30	5.55	16.62
+DiffAug	9.61	4.43	14.60	9.51	4.71	14.83
+MC	9.66	4.51	14.71	9.62	4.61	14.82
+MC+DiffAug	**9.96**	**3.61**	**13.60**	**9.94**	**3.93**	**13.72**
StyleGAN2	-	-	-	10.17	3.7	14.04
+DiffAug	10.19	2.25	11.40	10.03	2.44	11.62
+MC+DiffAug	**10.26**	**2.16**	**11.04**	**10.10**	**2.36**	**11.30**

Table 2: Quantitative results of image generation on CIFAR-10 using BigGAN/StyleGAN2 w/o and with our MC method and Differentiable Augmentation.

Improved Recognizability Metrics We generated 10k (the same setting as the test set) images using the BigGAN backbone. The WA rates are 62.56%, 52.09%, and 54.71% for the real test set, the generated set from Hinge baseline, and the generated set from Hinge + MC, respectively. Our MC method's images perform closer to the real test set than the baseline's, showing better distribution matching to the real data in terms of recognizability. The SA rate of our MC method is 83.42% compared to 93.65% of the real test set, showing that we generate fairly recognizable fake images.

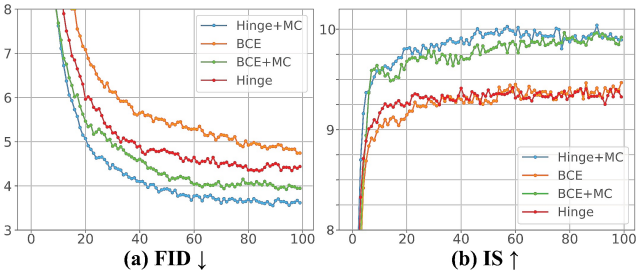


Figure 2: Learning curves in terms of (a) Fréchet Inception Distance and (b) Inception Score along the training on the CIFAR-10 using BigGAN with various loss combinations.

Qualitative Results The qualitative results are shown in Figure 3 and a figure in Appendix with only a small amount of images (in red boxes) misclassified by our classifier.

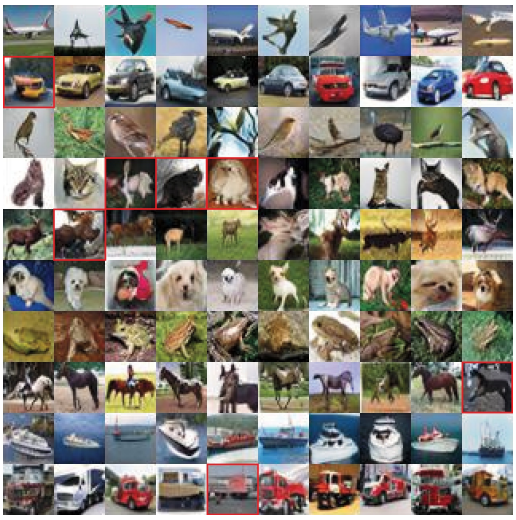


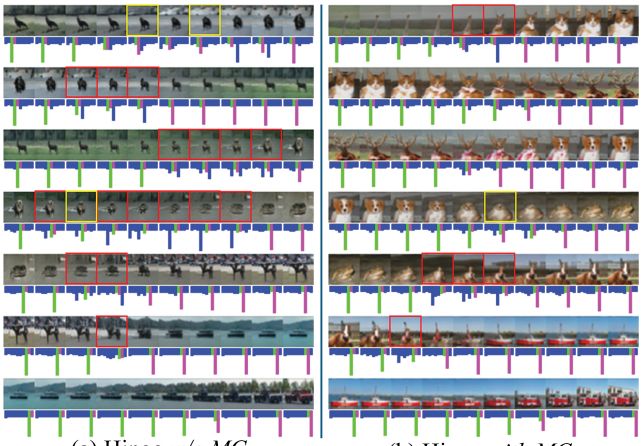
Figure 3: CIFAR-10 samples generated by the BigGAN backbone trained via Hinge + DiffAug + MC. Images in each row belong to one of the 10 classes. Images misclassified by ResNet-50 are in red boxes.

Latent Space Analysis The latent space learned by the generator is expected to be continuous and smooth so that small perturbations on the conditional input can lead to smooth and meaningful modifications on the generated output. To explore the latent space, we interpolate between each pair of randomly generated images by linearly interpolating their conditional inputs. The results are shown in Figure 4. Intermediary images between a pair of images from two different classes are shown in each row with their confidence score distributions below. The labels of the two classes are shown on the left and right sides of each row, respectively. Each distribution of the confidence scores is calculated by the bottleneck representation of the ResNet-50 classifier with a softened softmax function of tempera-

Loss	Hinge			BCE		
Metrics	IS ↑	FID ↓	IFID ↓	IS ↑	FID ↓	IFID ↓
BigGAN	10.73	8.31	83.36	10.81	8.37	81.89
+DiffAug	10.72	7.37	80.00	10.71	7.61	80.48
+MC	11.39	6.97	80.20	11.59	6.99	80.91
+MC+DiffAug	**11.81**	**5.77**	**76.26**	**11.90**	**5.85**	**77.33**

Table 3: Quantitative results of image generation on CIFAR-100 using BigGAN w/o and with our MC method and Differentiable Augmentation.

ture 5.0 for normalization. The score bars of the left class and the right class are shown in green and magenta, respectively. The red boxes highlight the images being classified as a third class, while the yellow boxes mark images with non-monotonic confidence score transitions compared to their adjacent images. In other words, images in both red and yellow boxes are undesirable as they imply that the latent space is less continuous and less smooth. Comparing Figure 4a and 4b, the MC method performs better in the learned latent space, with most decision switches between classes occurring in the mid-range of interpolation.



(a) Hinge *w/o MC* (b) Hinge *with MC*

Figure 4: Latent space interpolation based on cStyleGAN2 backbone trained via Hinge loss w/o and with our MC method. Red and yellow boxes highlight two types of undesirable transitions between generated images.

Quantitative Results on CIFAR-100 For completeness, we show the image generation performance on CIFAR-100 in Table 3. Significant improvements are achieved by using our MC method independently for both baseline discriminative losses, with an average improvement of 1.1 in IS, 1.6 in FID, and 3.7 in IFID. A detailed sensitive analysis w.r.t the Monte Carlo sample size is provided in the appendix.

Large-Scale and High-Resolution Dataset Results For the FFHQ256 (high-resolution), the lmageNet64 (large-scale), and the LSUN bedroom (large-scale and high-resolution) dataset, we use the StyleGAN2-ada (Karras et al. 2020a) as backbones. As shown in Table 4[2], MCGAN achieved significant and consistent gains in both FID and IS, as evidenced by 16.4% (4.51 → 3.77), 15.5% (19.83 → 16.76), and 35.7% (4.34 → 2.79) FID improvement, respectively, on FFHQ256, ImageNet64, and LSUN bedroom datasets. These improvements are significant and consistent during training periods and across various datasets, demonstrating faster convergence and better generation ability.

Dataset	Method	FID ↓	IS ↑	Precision↑	Recall ↑
FFHQ256	original	4.51	5.10	**0.69**	0.40
	+MC	**3.77**	**5.25**	0.69	**0.45**
ImageNet64	original	19.83	13.67	**0.65**	0.33
	+MC	**16.76**	**13.96**	0.63	**0.43**
LSUN bedroom	original	4.34	2.45	0.57	0.22
	+MC	**2.79**	**2.45**	**0.61**	**0.23**

Table 4: Quantitative results of image generation on large-scale and high-resolution datasets using StyleGAN2-ada w/o and with our MC method; FID is 10-run average.

Conditional Video Generation

The conditional video generation task aims to generate the next frame given the past frames of the videos. Here, we used the Moving MNIST data set (Srivastava, Mansimov, and Salakhudinov 2015), which consists of 10,000 20-frame 64x64 videos of moving digits. The whole dataset is divided into the training set (9,000 samples) and the test set (1,000 samples). For the architecture of both the generator and discriminator, we use the convolutional LSTM (ConvLSTM) unit proposed by (Shi et al. 2015) due to its effectiveness in video prediction tasks. In the model training, the generator takes in 5 past frames as the input and generates the corresponding 1-step future frame, then the real past frames and the generated future frames are concatenated along time dimension and put into the discriminator.

For comparison, we used classical GAN as the benchmark. We trained our model for 20,000 epochs with batch size 16. The model performance is evaluated by computing the MSE between the generated frames and the corresponding ground truth on the test set. Numerical results show that our proposed MC method reduces GAN's MSE from 0.1012 to 0.0840. Compared to the baseline, the predicted frames from our MC method are clearer, more coherent, and visually closer to the ground truth, as shown in Figure 5.

Conditional Time-Series Generation

Following (Liao et al. 2024), we consider the conditional time-series generation task on two types of datasets (1) d-dimensional vector auto-regressive (VAR) data and (2) empirical stock data. The goal is to generate 3-step future paths based on the 3-lagged value of time series. We apply the MCGAN to the RCGAN baseline (Esteban, Hyland, and Rätsch 2017) and benchmark it with TimeGAN (Yoon, Jarrett, and Van der Schaar 2019), GMMN (Li, Swersky,

[2]Baseline results differ from StyleGAN2-ADA's official benchmarks due to hyperparameter adjustments for different GPU setups.

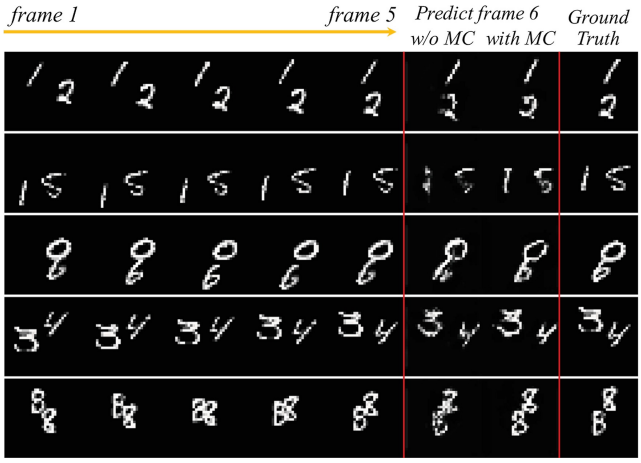


Figure 5: Results of predicting the next frame given the past 5 frames using ConvLSTM w/o and with our MC method.

and Zemel 2015) and SigWGAN (Liao et al. 2024) as the strong SOTA models for conditional time series generation. The model performance is evaluated using metrics in (Liao et al. 2024) including (1) ABS metric, (2) Correlation metric, (3)ACF metric and (4) R^2 error to assess the fitting of synthetic data in terms of marginal distribution, correlation, autocorrelation and usefulness, respectively.

VAR Dataset To validate MCGAN for multivariate time series systematically, we use VAR datasets with various path dimensions $d \in [1, 100]$ and various parameter settings. For $d \in \{1, 2, 3\}$, MCGAN consistently outperforms the RCGAN and TimeGAN (see results in Appendix). Figure 6 shows that the MCGAN and SigCWGAN have better fitting than other baselines in terms of conditional law as the estimated mean is closer to that of the ground truth compared with the others for $d = 3$. Note that SigCWGAN suffers the curse of dimensionality resulting from large d and becomes infeasible for $d \geq 50$, whereas MCGAN does not. In fact, as shown in Table 5, as d increases, the performance gains of MCGAN become more pronounced. With $d = 100$, the MC method improves all the metrics by 30%-40%, further highlighting its effectiveness in high-dimensional settings.

Stock Dataset The stock dataset is a 4-dimensional time series composed of the log return and log volatility data of S&P 500 and DJI spanning from 2005/01/01 to 2020/01/01. To cover the stylized facts of financial time series like leverage effect and volatility clustering, we also evaluate our generated samples using the ACF metric on the absolute return and squared return. Table 6 demonstrates that our MC method consistently improves the generator performance in terms of temporal dependency, cross-correlation and usefulness. Although RCGAN achieved comparable ABS metrics, it failed to capture the cross-correlation and temporal dependence. Specifically, using our proposed MC method, the correlation metric and ACF metric of RCGAN can be improved from 0.25184 to 0.15687 and from 0.03814 to 0.02905. The gap in the R^2 further showcases that our MC method can enhance the generator to generate high-fidelity samples.

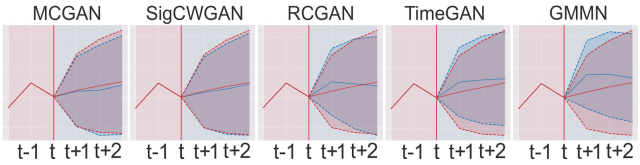


Figure 6: Comparison of models' performance in fitting the conditional distribution of future time series given one past path sample. The real and generated paths are plotted in red and blue, respectively, with the shaded area as the 95% confidence interval. The synthesized data is VAR(1) for $d = 3$.

Loss		Hinge			BCE		
d	Method	ABS ↓	Corr ↓	ACF ↓	ABS ↓	Corr ↓	ACF ↓
10	RCGAN	0.0180	0.0568	0.0818	0.0153	0.0507	0.0900
	+MC	**0.0155**	**0.0436**	**0.0651**	**0.0139**	**0.0459**	**0.0719**
50	RCGAN	0.0353	0.0700	0.0884	0.0363	0.0710	0.0877
	+MC	**0.0286**	**0.0616**	**0.0687**	**0.0250**	**0.0600**	**0.0700**
100	RCGAN	0.0332	0.0790	0.1102	0.0379	0.0730	0.1022
	+MC	**0.0230**	**0.0498**	**0.0669**	**0.0234**	**0.0502**	**0.0614**

Table 5: Quantitative results of time-series generation on VAR data with different path dimensions d ranging from 10 to 100 using RCGAN w/o and with our MC method.

Model	ABS ↓	ACF ↓	ACF($\lvert x\rvert$) ↓	ACF(x^2) ↓	Corr ↓	R^2 (%) ↓
RCGAN	**0.0087**	0.0381	0.0788	0.1393	0.2518	4.4968
MCGAN (ours)	0.0100	**0.0291**	**0.0544**	0.0993	0.1569	**2.8429**
SigCWGAN	0.0096	0.0298	0.1339	**0.0846**	**0.1172**	3.8198
GMMN	0.0139	0.0599	0.2530	0.2696	0.3184	11.8758
TimeGAN	0.0110	0.0572	0.0690	0.1258	0.4734	4.5396

Table 6: Quantitative results of time-series generation on SPX/DJI data using RNN w/o and with our MC method.

Conclusion

This paper presents a general MCGAN method to tackle the training instability, a key bottleneck of GANs. Our method enhances generator training by introducing a novel regression loss for (conditional) GANs. We establish the optimality and discriminability of MCGAN, and prove that the convergence of optimal generator can be achieved under a weaker condition of the discriminator due to the strong supervision of the regression loss. Moreover, extensive numerical results on various datasets, including image, time series data, and video data, are provided to validate the effectiveness and flexibility of our proposed MCGAN and consistent improvements over the benchmarking GAN models.

For future work, it is worthwhile to explore the application of MCGAN to enhance state-of-the-art GAN models for more challenging and complex tasks, such as text-to-image generation. Besides, given the flexibility and promising results of the MCGAN on different types of data, it can be effectively applied to generate multi-modality datasets simultaneously. Moreover, MCGAN can be extended to incorporate more advanced discriminative losses, than those used in our numerical study, for further performance improvement.

Acknowledgements

HN and WY are supported by the EPSRC under the program grant EP/S026347/1 and the Alan Turing Institute under the EPSRC grant EP/N510129/1. WY is also supported by the Data Centric Engineering Programme (under the Lloyd's Register Foundation, UK grant G0095). The authors are grateful to Richard Hoyle from UCL, Terry Lyons from Oxford, and the Oxford Mathematical Institute IT staff for their support with computing resources. We also thank Lei Jiang for his help with some of the numerical experiments and the anonymous referees for their constructive suggestions, which significantly improved the paper.

References

Alex, K. 2009. Learning multiple layers of features from tiny images. *https://www. cs. toronto. edu/kriz/learning-features-2009-TR. pdf*.

Arjovsky, M.; and Bottou, L. 2017. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, 214–223. PMLR.

Brock, A.; Donahue, J.; and Simonyan, K. 2018. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.

Esteban, C.; Hyland, S. L.; and Rätsch, G. 2017. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30.

Gupta, S.; Keshari, A.; and Das, S. 2022. Rv-gan: Recurrent gan for unconditional video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024–2033.

Han, C.; Hayashi, H.; Rundo, L.; Araki, R.; Shimoda, W.; Muramatsu, S.; Furukawa, Y.; Mauri, G.; and Nakayama, H. 2018. GAN-based synthetic brain MR image generation. In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, 734–738. IEEE.

Hou, L.; Cao, Q.; Shen, H.; Pan, S.; Li, X.; and Cheng, X. 2022. Conditional gans with auxiliary discriminative classifier. In *International Conference on Machine Learning*, 8888–8902. PMLR.

Kang, M.; Shim, W.; Cho, M.; and Park, J. 2021. Rebooting acgan: Auxiliary classifier gans with stable training. *Advances in Neural Information Processing Systems*, 34: 23505–23518.

Kang, M.; Shin, J.; and Park, J. 2022. StudioGAN: A Taxonomy and Benchmark of GANs for Image Synthesis. *arXiv preprint arXiv:2206.09479*.

Karras, T.; Aittala, M.; Hellsten, J.; Laine, S.; Lehtinen, J.; and Aila, T. 2020a. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*, 33: 12104–12114.

Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2020b. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8110–8119.

Li, Y.; Swersky, K.; and Zemel, R. 2015. Generative moment matching networks. In *International conference on machine learning*, 1718–1727. PMLR.

Liao, S.; Ni, H.; Sabate-Vidales, M.; Szpruch, L.; Wiese, M.; and Xiao, B. 2024. Sig-Wasserstein GANs for conditional time series generation. *Mathematical Finance*, 34(2): 622–670.

Lim, J. H.; and Ye, J. C. 2017. Geometric gan. *arXiv preprint arXiv:1705.02894*.

Liu, S.; Bousquet, O.; and Chaudhuri, K. 2017. Approximation and convergence properties of generative adversarial learning. *Advances in Neural Information Processing Systems*, 30.

Mao, X.; Li, Q.; Xie, H.; Lau, R. Y.; Wang, Z.; and Paul Smolley, S. 2017. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 2794–2802.

Mescheder, L.; Geiger, A.; and Nowozin, S. 2018. Which training methods for GANs do actually converge? In *International conference on machine learning*, 3481–3490. PMLR.

Mirza, M.; and Osindero, S. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Miyato, T.; Kataoka, T.; Koyama, M.; and Yoshida, Y. 2018. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*.

Miyato, T.; and Koyama, M. 2018. cGANs with projection discriminator. *arXiv preprint arXiv:1802.05637*.

Ni, H.; Szpruch, L.; Sabate-Vidales, M.; Xiao, B.; Wiese, M.; and Liao, S. 2021. Sig-Wasserstein GANs for time series generation. In *Proceedings of the Second ACM International Conference on AI in Finance*, 1–8.

Odena, A.; Olah, C.; and Shlens, J. 2017. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*, 2642–2651. PMLR.

Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; and Woo, W.-c. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28.

Srivastava, N.; Mansimov, E.; and Salakhudinov, R. 2015. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, 843–852. PMLR.

Tseng, H.-Y.; Jiang, L.; Liu, C.; Yang, M.-H.; and Yang, W. 2021. Regularizing generative adversarial networks under limited data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7921–7931.

Xu, L.; Skoularidou, M.; Cuesta-Infante, A.; and Veeramachaneni, K. 2019. Modeling tabular data using conditional gan. *Advances in neural information processing systems*, 32.

Xu, T.; Wenliang, L. K.; Munn, M.; and Acciaio, B. 2020. Cot-gan: Generating sequential data via causal optimal transport. *Advances in neural information processing systems*, 33: 8798–8809.

Yoon, J.; Jarrett, D.; and Van der Schaar, M. 2019. Time-series generative adversarial networks. *Advances in neural information processing systems*, 32.

Zhao, J. 2016. Energy-based Generative Adversarial Network. *arXiv preprint arXiv:1609.03126*.

Zhao, S.; Liu, Z.; Lin, J.; Zhu, J.-Y.; and Han, S. 2020. Differentiable augmentation for data-efficient gan training. *Advances in Neural Information Processing Systems*, 33: 7559–7570.

Zhou, P.; Xie, L.; Ni, B.; Geng, C.; and Tian, Q. 2021. Omni-gan: On the secrets of cgans and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14061–14071.