

Consistency-GAN: Training GANs with Consistency Model

Yunpeng Wang¹, Meng Pang^{1*}, Shengbo Chen^{2*}, Hong Rao^{1*}

¹ Nanchang University

² Henan University

408000220012@email.ncu.edu.cn, pangmeng1992@gmail.com, ccb02kingdom@gmail.com, raohong@ncu.edu.cn

Abstract

For generative learning tasks, there are three crucial criteria for generating samples from the models: quality, coverage/diversity, and sampling speed. Among the existing generative models, Generative adversarial networks (GANs) and diffusion models demonstrate outstanding quality performance while suffering from notable limitations. GANs can generate high-quality results and enable fast sampling, their drawbacks, however, lie in the limited diversity of the generated samples. On the other hand, diffusion models excel at generating high-quality results with a commendable diversity. Yet, its iterative generation process necessitates hundreds to thousands of sampling steps, leading to slow speeds that are impractical for real-time scenarios. To address the aforementioned problem, this paper proposes a novel Consistency-GAN model. In particular, to aid in the training of the GAN, we introduce instance noise, which employs consistency models using only a few steps compared to the conventional diffusion process. Our evaluations on various datasets indicate that our approach significantly accelerates sampling speeds compared to traditional diffusion models, while preserving sample quality and diversity. Furthermore, our approach also has better model coverage than traditional adversarial training methods.

Introduction

Generative models are evaluated based on three primary performance indicators: *sampling quality*, *diversity*, and *sampling speed of the generated samples*. Upon their introduction, GANs (Goodfellow et al. 2014; Radford, Metz, and Chintala 2016; Brock, Donahue, and Simonyan 2018; Wu et al. 2019; Karras, Laine, and Aila 2019; Karras et al. 2020; Pang et al. 2021a,b; Sauer et al. 2021; Sauer, Schwarz, and Geiger 2022) demonstrate remarkable capabilities in generating realistic and high-quality samples. However, GANs-generated samples often lack diversity, indicating that the generator network tends to produce a limited range of patterns from the data distribution during training, thereby failing to cover the entire data space. This phenomenon is commonly referred to as the “model collapse” (Kodali et al. 2017) problem. To address this issue, Salimans et al. (Salimans et al. 2016) propose a method of injecting noise in

Model	High quality	Fast sampling	High diversity
GANs	✓	✓	✗
DMs	✓	✗	✓
Ours	✓	✓	✓

Table 1: Comparison of GANs, diffusion models, and our proposed method in terms of three essential criteria for desired sample generation.

the process of training GANs. This approach introduces a certain level of randomness in the input of the generator network, enabling the generation of more diverse samples. Nevertheless, noise injection complicates the adversarial training between the generator and the discriminator, making it more challenging for them to estimate the gradient during backpropagation when updating parameters. Consequently, this can lead to gradient disappearance, gradient explosion, and other related problems, ultimately affecting the model’s stability and the effectiveness of the training process.

Diffusion models (DMs) (Ho, Jain, and Abbeel 2020; Song et al. 2020; Song, Meng, and Ermon 2020; Karras et al. 2022) generate samples by reversing the diffusion process. In comparison to GANs, diffusion models exhibit significant improvements in diversity while maintaining comparable sample quality. The reverse diffusion process empowers the diffusion models to generate samples with a wide range of diversity from initial noisy samples. However, due to the reliance on an iterative generative process, diffusion models suffer from slow sampling speed, which limits their applicability in real-time and interactive scenarios. To tackle this issue, a few attempts have recently been made. For instance, Song et al. (Song, Meng, and Ermon 2020; Song et al. 2022b,a) propose to expedite the sampling process by extending the diffusion model to encompass non-Markovian diffusion processes, while the authors (Wang et al. 2022) incorporate the forward diffusion process to introduce Gaussian mixed-distributed instance noise into the adversarial network. Although these above-mentioned methods can better capture the diversity of samples, the iterative generation process still imposes limitations on sampling efficiency, and cannot completely eliminate the limitation of slow sampling speed of the traditional diffusion models.

In light of the above, we propose a new adversarial train-

*Corresponding Author

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

ing network, dubbed Consistency-GAN, which is able to ensure high sample quality, high diversity, and fast sampling speed simultaneously in sample generation. As shown in Table 1, the proposed Consistency-GAN could address the limitation of poor coverage (low diversity) in GANs, as well as the slow sampling speed in diffusion models. In Consistency-GAN, a novel consistency mapping module is introduced into the generative adversarial network, which enables the instance noise to be injected into the adversarial training process efficiently and safely, thus striking a balance between model stability and sample diversity. Furthermore, the consistency mapping module is capable of reversibly mapping points along the diffusion trajectory of the probability flow ordinary differential equation (ODE) (Song et al. 2020) back to their origins, which could significantly reduce the time and computational resources required for learning diffusion trajectories through iterative generation processes.

In summary, the contributions of this paper are presented as follows:

- A new adversarial training network, i.e., Consistency-GAN, is proposed, which is the first work concurrently satisfying the three essential criteria for desired sample generation, i.e., high sample quality, high diversity, and fast sampling speed. Consistency-GAN addresses the limitation of low diversity in GANs, as well as the slow sampling speed in diffusion models.
- A novel consistency mapping module is designed, which efficiently injects instance noise into real and generated data distributions to improve sample diversity. Furthermore, a conditional discriminator dependent on the mapping step is developed to fit the consistency mapping module, which differentiates the original data distribution and the generated data distribution perturbed by noise.
- Qualitative and quantitative experiments are conducted on multiple real-world image datasets, which demonstrate the superior performance of the proposed method over the state-of-the-art GANs and diffusion models, in terms of sample quality, diversity, and sampling speed.

Related Works

Generative Adversarial Networks

GANs represent a class of generative models specifically designed to learn the underlying distribution of data. This is accomplished through a min-max game between generator and discriminator, thereby facilitating the target data distribution. The generator component takes a random noise vector as input and endeavors to produce synthetic samples that closely resemble the target data. Conversely, the discriminator component receives real data samples and fake samples generated by the generator, with the purpose of accurately discriminating between them. The two components compete against each other such that the generator strives for generating realistic-looking samples that fool the discriminator.

Recently, some GANs have attempted to introduce noise or penalty terms into the generator and discriminator, in order to regularize the training and mitigate the overfitting

problem. Salimans *et al.* (Salimans et al. 2016) weaken the discriminator by introducing label noise, i.e., randomly flipping training labels, so as to perturb the optimal decision boundary of the generator. Kodali *et al.* (Kodali et al. 2017) incorporate an additional gradient penalty term into the objective of the discriminator to restrain its optimization speed during training, thus stabilizing the model training. Karras *et al.* (Karras et al. 2017; Karras, Laine, and Aila 2019; Karras et al. 2020) add random noise progressively at different layers of the generator, which aims to prevent the training of the generator from converging to local optima. However, the random noise added by these methods can concurrently introduce a level of uncertainty in the resulting outputs. This lack of predictability may prove unfavorable in certain application scenarios where users seek meticulous command over the consistency of the generated outcomes. By contrast, our consistency mapping module can avoid this problem by directly adding instance noise to the generated samples during the adversarial training process.

Diffusion Models

Diffusion models (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2020; Kingma et al. 2023; Yang et al. 2023; Meng and Kabashima 2023) consists of two main components: the forward diffusion process and the reverse denoising process. The forward diffusion process involves the systematic addition of Gaussian noise to the data, progressively transforming it into random noise. In this process, numerous diffusion steps are performed with Gaussian noise continuously added to the original data. The forward diffusion process can be conceptualized as a Markov chain. Whereas, the reverse denoising process aims to gradually restore the original data distribution from a standard Gaussian distribution.

A major drawback of diffusion models lies in the slow sampling speed due to the large number of iterative sampling steps. To alleviate this issue, some attempts have recently been made, including performing knowledge distillation (Luhman and Luhman 2021), learning adaptive noise plan (San-Roman, Nachmani, and Wolf 2021; Pei et al. 2020, 2022), introducing non-Markov diffusion process (Song, Meng, and Ermon 2020; Kong and Ping 2021), and using a better stochastic differential equation solver (SDE) (Song and Ermon 2019; Jolicoeur-Martineau et al. 2020; Meng et al. 2022; Song et al. 2023) for continuous-time models. Lu *et al.* (Lu et al. 2022) introduce a fast training-free solver (Song, Meng, and Ermon 2020; Jolicoeur-Martineau et al. 2021; Bao et al. 2022) of diffusion ordinary differential equations (ODEs) (Song et al. 2020; Karras et al. 2022) for fast sampling by leveraging the semi-linearity of diffusion ODEs. Most recently, Wang *et al.* (Wang et al. 2022) propose a novel Diffusion-GAN model, which utilizes the diffusion chain to generate Gaussian mixed distributed instance noise and adds it to the real and generated data distributions during training, so as to stabilize the training process. Although this new way of adding noise mitigates the discriminator over-fitting, it still cannot completely eliminate the limitation of slow sampling speed in Diffusion-GAN due to the iterative generation process.

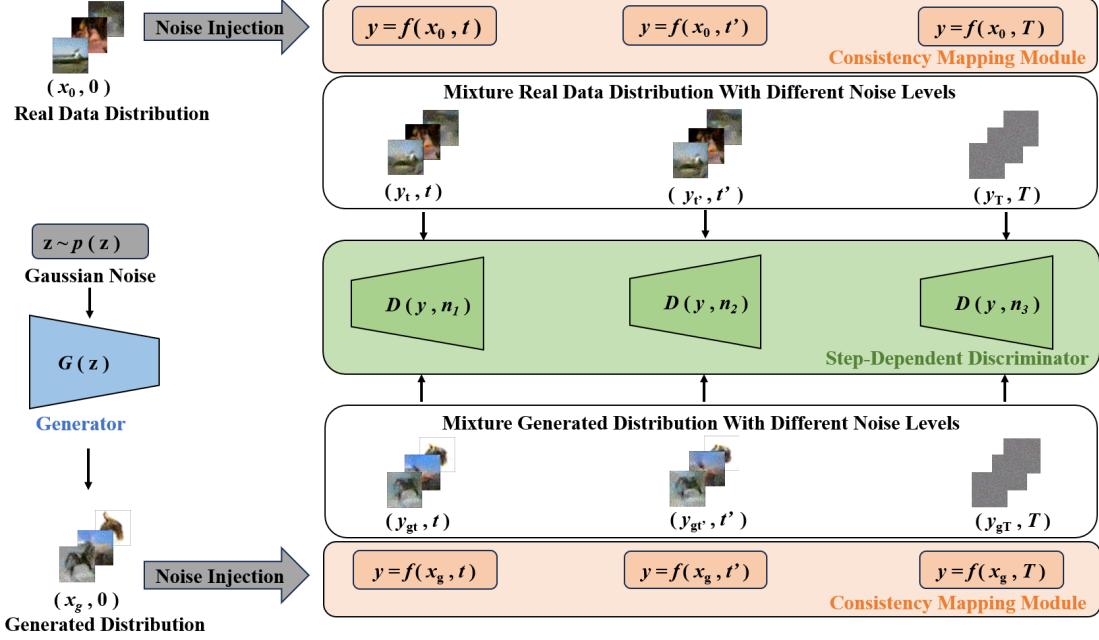


Figure 1: The flowchart of Consistency-GAN.

The Proposed Method

As illustrated in Figure 1, the proposed Consistency-GAN model consists of three key modules: the consistency mapping module, the generator module, and the step-dependent discriminator module. Specifically, the first module is used to add instance noise into both the real and generated data distributions, while the latter two modules play an adversarial game to generate samples of the target distribution. The training of Consistency-GAN involves two steps, i.e., consistency mapping training and adversarial training, which are presented in **Algorithm 1**.

Consistency Mapping Training

To achieve a more robust and diverse sample generation task, we introduce a novel consistency mapping module into the generative adversarial network, which enables the instance noise to be injected into the adversarial training process efficiently and safely.

The training process of the consistency mapping module corresponds to Step I in Algorithm 1. Specifically, the consistency mapping module introduces the SDE (Song et al. 2020; Karras et al. 2022) to simulate the diffusion process of original data distribution $p(x)$ in continuous-time diffusion models. Furthermore, it utilizes a score-based model (Song et al. 2021) to compute the gradient field of data distributions at each mapping step, thus facilitating effective noise injection. Then, we use probability flow ODE to solve the reverse-time SDE, providing trajectories for the data distribution $p_t(x)$ at a given moment t ,

$$dx_t = [g(x, t) - \frac{1}{2}h(t)^2 \nabla \log p_t(x)]dt, \quad (1)$$

where $g(x, t)$ is drift coefficient of x_t , $h(t)$ is diffusion coefficient of x_t , $\nabla \log p_t(x)$ is the score function of $p_t(x)$.

For any point x_t on the trajectory of probability flow ODE, our target is to train a function $f(\cdot)$ mapping points on the same probability flow ODE to the original point, which can be written as $f(x_t, t) = x_0$. We introduce parameter k to divide the trajectory into $k + 1$ time points t_1, t_2, \dots, t_{k+1} , in each of which we minimize $L(\cdot)$ to ensure the strong consistency of our module function $f(\cdot)$. Parameterizing it with a neural network as $f_\gamma(x_t, t)$, the loss function for training the consistency module is given as follows:

$$L(\gamma, \gamma^-) = \mathbb{E}[d(f_\gamma(x_{k+1} + t_{k+1}z, t_{k+1}), f_{\gamma^-}(x_k + t_kz, t_k))], \quad (2)$$

where $z \sim N(0, I)$ is the standard Gaussian noise, $k \sim U[1, T - 1]$ denotes the uniform distribution. $x \sim p(x)$ is the initial sample, $x_k \sim N(x, t^2 I)$, $d(\cdot, \cdot)$ represents the distance metric function, γ^- denotes the updated value of parameter γ after each optimization step, and $\mathbb{E}(\cdot)$ denotes the expectation over all relevant random variables. Note that the mapping function f_γ is differentiable, allowing us to minimize the objective function $L(\gamma, \gamma^-)$ through stochastic gradient descent on the parameter γ . Given a decay rate μ , we update the parameter γ^- using exponential moving average (EMA) as follows:

$$\gamma^- \leftarrow \mu\gamma^- + (1 - \mu)\gamma. \quad (3)$$

Subsequently, we use the well-trained consistency mapping module to inject instance noise into the generated sample x_g . The mapping process starts from the original sample x and the generated sample x_g , and reaches the desired noise level after N steps. Samples from the consistency mapping module can be partitioned based on the current mapping steps n , and the mixture distribution y represents noisy versions of real samples and generated samples.

Algorithm 1: Training of Consistency-GAN

Input: Random noise $z \sim p(z)$, Original data $x \sim p(x)$
Parameter: Initial training parameter θ and γ , Sequence of time points t_1, t_2, \dots, t_{k+1} , Inference step N
Output: Generated samples $x_g = G(z)$, Noise mixture samples $y = f_\gamma(x, t)$

- 1: **Step I: Consistency mapping training**
- 2: $k \leftarrow 0$
- 3: **repeat**
- 4: Sample $x \sim p(x)$
- 5: Sample $z \sim N(0, I)$
- 6: Update γ using Equation (2) and (3)
- 7: $k \leftarrow k + 1$
- 8: **until** convergence
- 9: **Step II: Adversarial training**
- 10: **while** $i \leq$ number of training iterations **do**
- 11: Sample noise samples: $z \sim p(z)$.
- 12: Sample from original data: $x \sim p(x)$.
- 13: Obtain generated samples: $x_g \sim G(z)$.
- 14: Obtain pretrained consistency module $f_\gamma(\cdot)$.
- 15: Noise injection:
- 16: **for** $j = 1$ to N **do**
- 17: $y \leftarrow f(x, j)$
- 18: $y_g \leftarrow f(x_g, j)$
- 19: **end for**
- 20: Update θ in Equation (4).
- 21: **end while**

Adversarial Training

Before the introduction of the adversarial training objective of the proposed Consistency-GAN, we briefly review the adversarial training process of traditional GANs to facilitate understanding. Given the original data distribution $p(x)$ and a simple prior distribution $p(z)$, random noise z is sampled from the prior distribution and inputted into the generator G to obtain generated samples $G(z)$. Real data samples x extracted from the original distribution and $G(z)$ produced by the generator are inputted into the discriminator. The objective of adversarial training is to enable the discriminator D to correctly discriminate whether the input comes from $p(x)$ or $G(z)$, while simultaneously allowing the generator to generate samples that can deceive the discriminator.

In comparison to traditional GANs, the adversarial training objective of Consistency-GAN is to train a discriminator D that is dependent on the number of mapping steps n , and can distinguish between real and generated noisy perturbed samples. Additionally, we also aim to train a generator G that is capable of fitting the real noise sample distribution to the generated noise sample distribution. The adversarial training process with the inclusion of the mapping module corresponds to Step II in Algorithm 1.

Formally, the objective functions for adversarial training are defined by the following equation,

$$V(G, D) = \mathbb{E}_{x \sim p(x), y \sim f(x)} [\log D(y, n)] + \mathbb{E}_{y_g \sim f(G_\theta(z))} [\log(1 - D(y_g, n))], \quad (4)$$

where $p(x)$ represents the true data distribution, $n \sim$

$U[1, N - 1]$ denotes uniform distribution, and $y \sim f(x)$ denotes our consistency mapping module. The objective function encourages the discriminator to classify noisy perturbed samples from the original data as real and those from the generated data as fake. The generator, on the other hand, strives to generate samples $y_g \sim f(G_\theta(z))$ that can deceive the discriminator at each mapping step. The generator's differentiable parameters θ in the objective function can be optimized using gradient descent.

During the adversarial training process, random noise is input into the generator to generate new samples. The instance noise injected through the consistency mapping ensures that our discriminator converges in a more robust manner at each training step. The impact of this noise injection approach on training will be further analyzed in conjunction with the experimental results in the subsequent sections.

Experiments

Datasets

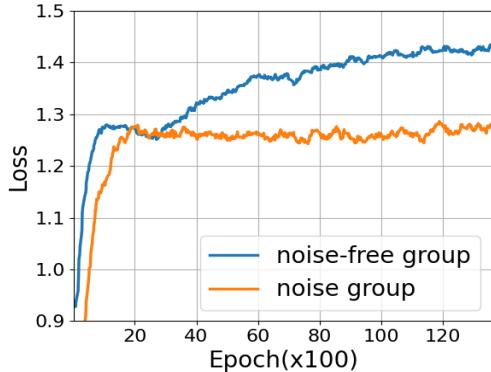
We conduct a series of experiments on datasets ranging from low to high resolutions, including CIFAR-10 (32×32) (Krizhevsky et al. 2009), STL-10 (64×64) (Coates, Ng, and Lee 2011) and CelebA (256×256) (Liu et al. 2015). The CIFAR-10 dataset consists of 50,000 training images with a resolution of 32×32 , while the STL-10 dataset contains 100,000 images with a resolution of 96×96 . During the data preprocessing stage, we centrally crop the STL-10 images to a resolution of 64×64 . We also partition the CelebA dataset based on resolution. Subsequently, we select a subset of 30,000 images of resolutions 256×256 for training the model.

Parameter Setting

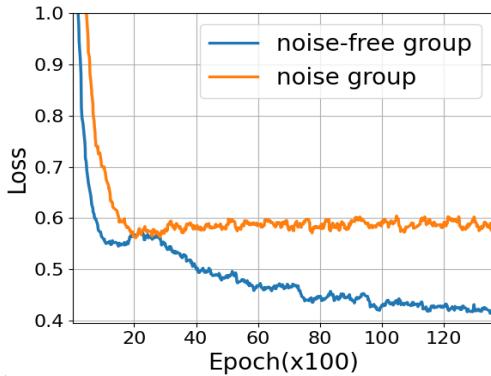
We implement Consistency-GAN using PyTorch. During training, we utilize a consistent mapping module to map the features obtained from a pre-trained feature network and feed them back into the conditional discriminator based on the current mapping step n for training. The parameters for training the consistency mapping module are empirically set as follows: $\mu_0 = 0.9$ (for consistency regularization weight), distance metric function d using the L_2 metric. During adversarial training, the consistency mapping is performed for $N = 5$ steps, and Gaussian noise with a standard deviation of $\sigma = 0.5$ is added. The training is conducted using 4 NVIDIA A800 GPUs, with a batch size of 64 and a total of 20,000 training iterations.

Evaluation Protocol

To assess the quality of the generated samples, we employ commonly used metrics, including Inception Score (IS) (Salimans et al. 2016) and Fréchet Inception Distance (FID) (Heusel et al. 2017; Bińkowski et al. 2018) to evaluate their fidelity to the real data distribution, where higher IS scores and lower FID scores indicate better fidelity. For the calculation of metrics, we compare 50,000 generated samples from our model against the 50,000 images in the CIFAR-10 training set. In addition, we utilize Recall Score (Sajjadi



(a) Discriminator loss



(b) Generator loss

Figure 2: The impact of introducing instance noise.

Model	IS↑	FID↓	Recall↑
BigGAN	9.22	14.7	0.36
StyleGAN2	9.18	11.1	0.41
TransGAN	9.05	9.26	0.40
Projected GAN	8.82	3.39	0.42
Consistency-GAN (ours)	9.76	3.16	0.47

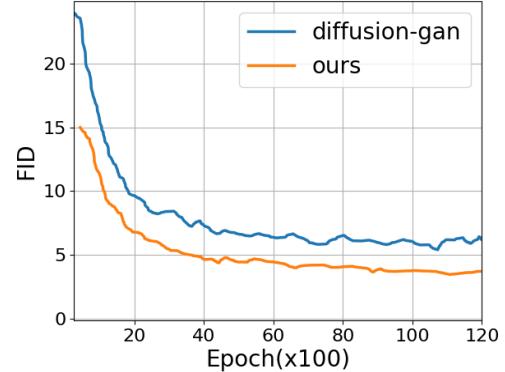
Table 2: The results on CIFAR-10 compared with GANs.

et al. 2018; Kynkänniemi et al. 2019) to evaluate the sample diversity, where a higher recall score indicates greater diversity in the generated samples. As for sampling speed, a widely recognized metric is the Number of Function Evaluations (NFE).

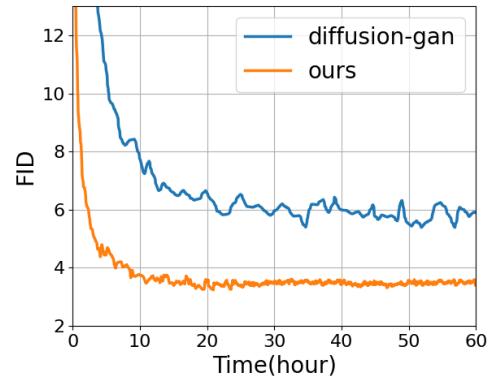
Comparison with GANs and Diffusion Models

We conduct a comprehensive comparison and analysis between our method and other existing methods including both GANs and diffusion models, on the CIFAR-10 dataset,

In Table 2, we present the quantitative comparison results of our method against a bunch of GANs, including BigGAN (Brock, Donahue, and Simonyan 2018), StyleGAN2 (Karras et al. 2020), TransGAN (Jiang, Chang, and Wang 2021), Projected GAN (Sauer et al. 2021). We observe that, compared to these GANs, our method has better IS and FID performance. In addition, our method also achieves a higher Re-



(a) Model performance



(b) Convergence speed

Figure 3: Model performance and convergence speed.

Model	IS↑	FID↓	NFE↓
DDPM	9.46	3.21	1000
DDIM	9.18	8.23	10
Diffusion-GAN	9.54	4.92	30
Consistency-GAN (ours)	9.76	3.16	5

Table 3: The results on CIFAR-10 compared with diffusion models.

call. Therefore, we claim that our method significantly improves the model diversity, as well as improving the sample quality, compared to other GANs. This observation validates the effectiveness of our consistency mapping module in enhancing diversity.

In Table 3, we compare our method with other diffusion models, including DDPM (Ho, Jain, and Abbeel 2020), DDIM (Song, Meng, and Ermon 2020) and the state-of-the-art Diffusion-GAN (Wang et al. 2022), in terms of sample quality and sampling efficiency. It is observed that, in contrast to the traditional diffusion models that require hundreds or thousands of steps for sampling, our method surpasses both DDPM and DDIM in terms of all three criteria. For the Diffusion-GAN, which allows for dynamic adjustment of the diffusion process time steps, our method achieves better IS and FID when the NFE of our Consistency-GAN is only 1/6 of that of Diffusion-GAN.

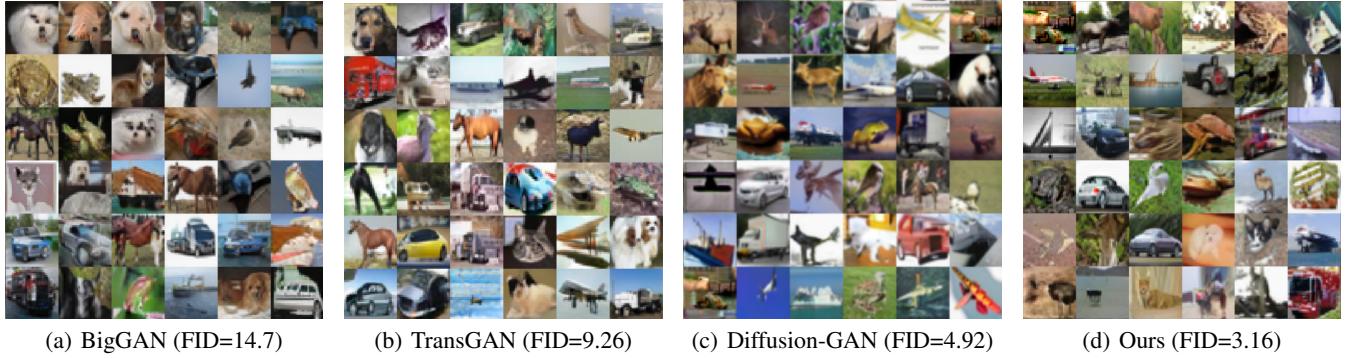


Figure 4: Sampling results on CIFAR-10 dataset.



Figure 5: Sampling results on STL-10 dataset.

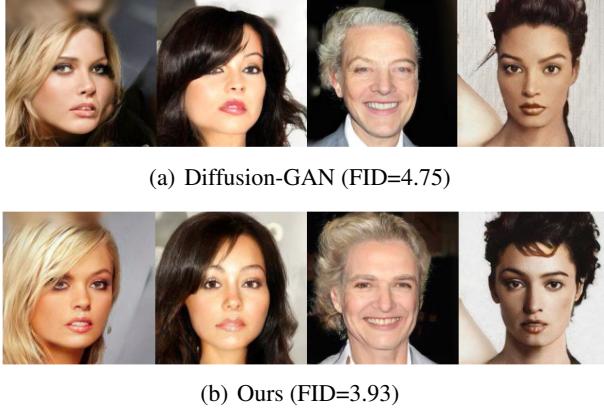


Figure 6: Sampling results on CelebA (256×256) dataset.

As depicted in Figure 2, we illustrate the impact of introducing instance noise during the adversarial training process on the loss curves of the generator and discriminator networks. The blue line represents the adversarial training process without any noise perturbation. Initially, the losses of both the generator and discriminator converge rapidly towards their respective target values. However, as the training progresses, the loss curves deviate from their target values, indicating a phenomenon known as mode collapse. This scenario hinders the generator from producing diverse samples and prevents the discriminator from accurately distinguishing between real and generated samples, ultimately hampering the network from reaching the desired performance level. On the other hand, the orange line represents the con-

trol group where instance noise is injected through the consistency mapping module during the training process. The injected noise increases the burden on the discriminator, leading to a slightly slower convergence speed. However, this enhances the model’s robustness against disturbances and allows it to focus more on the underlying data characteristics. In the later stages of training, the loss curves of the generator and discriminator fluctuate near their target values, demonstrating excellent model stability.

In Figure 3, we show the FID curves for both our approach and Diffusion-GAN. We can see that our model has a much faster convergence speed with respect to both steps and time. For example, to reach an FID of 5, Diffusion-GAN requires 20,000 iterations (more than 60 hours) while our Consistency-GAN only needs 3,000 steps (around 4 hours). Furthermore, our approach achieves a FID as low as 3.16.

In Figures 4-5, we present the qualitative sampling results associated with the FID values on the CIFAR-10 dataset (32×32) and STL-10 dataset (64×64). It can be observed that our Consistency-GAN model generates a wide variety of lifelike image samples and achieves better FID results compared to diffusion-GAN and other GANs, indicating that our model generates samples closer to the real ones. Furthermore, in Figure 6, we compare samples generated by our model and Diffusion-GAN using the same random seeds in the CelebA dataset with higher resolutions (256×256). We find that our model still achieves better and more stable performance than Diffusion-GAN in terms of face generation, with generated faces being less prone to deformity and appearing more realistic and natural. The above experiments validate the superiority of our model in generating images at different resolutions.

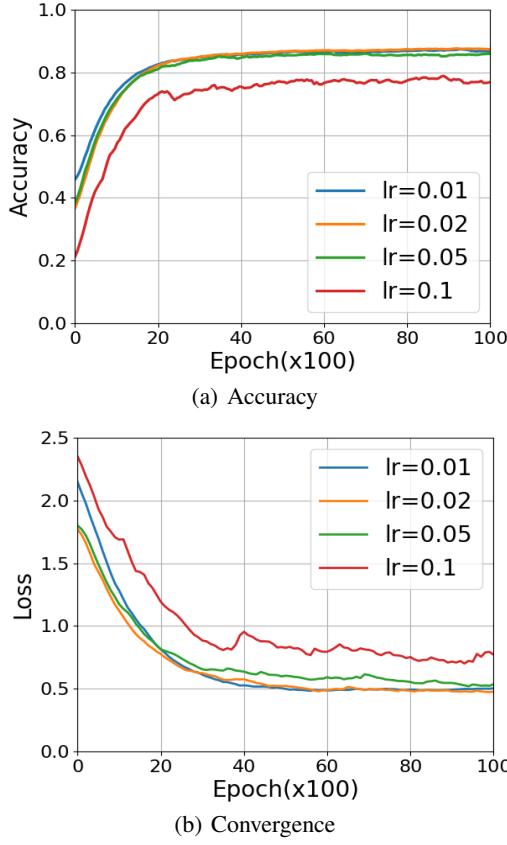


Figure 7: Accuracy and convergence speed under different learning rates.

Parameter Sensitivity Study

This subsection conducts a series of experiments on the CIFAR-10 dataset to investigate the impact of various parameters during the training process.

Initially, we study the impact of training the consistency mapping module with different learning rates on model accuracy and convergence speed. As shown in Figure 7 (a) and (b), setting the learning rate to 0.02 results in a more stable training process and maintains higher accuracy on the test set. The results in Figure 8 validate the convergence speed of training the consistency mapping module using different distance metrics: L_1 distance ($d(x, y) = \|x - y\|_1$), L_2 distance ($d(x, y) = \|x - y\|_2^2$), and LPIPS (Zheng et al. 2023). When selecting L_2 as the distance metric, the model exhibits a faster convergence speed.

We also explore the effects of using different numbers of mapping steps. From Figure 9 (a) and (b), we observe that when increasing the number of mapping steps, both the sample quality and diversity of the model improve. The best results in terms of FID and IS are achieved at $T = 5$, with a relatively high Recall as well. This confirms the advantages of incorporating multiple mapping steps during the adversarial training process, as it significantly enhances sample quality and diversity. However, increasing the number of mapping steps beyond a certain threshold may even lead to

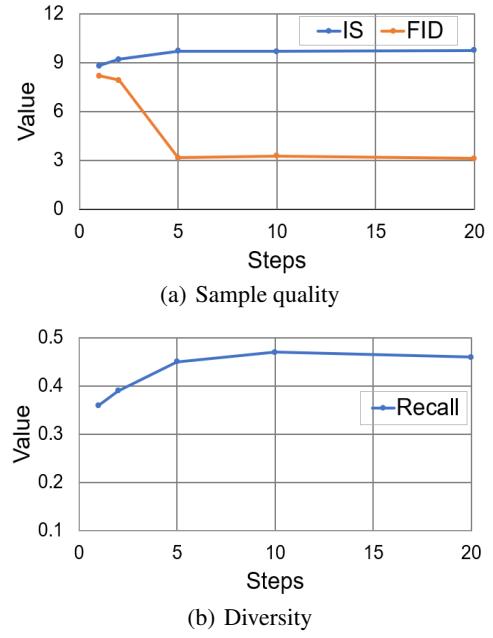


Figure 8: Performance under different mapping steps.

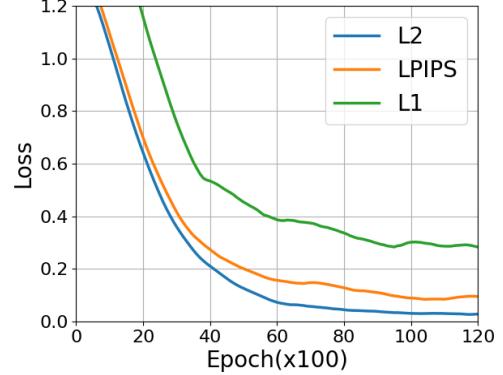


Figure 9: Convergence speed with different distance metrics.

a worse Recall performance.

Conclusion

In this paper, we propose the Consistency-GAN, which introduces an efficient approach to add instance noise to both the real and generated distributions of GANs through consistency mapping with only a few steps. Through experimental validation, our Consistency-GAN demonstrates comparable sample quality and diversity to the latest diffusion models, while achieving a significant improvement in sampling speed. Compared to traditional GANs, our model exhibits superior model coverage and sample diversity. The proposed Consistency-GAN in this paper largely overcomes the trilemma of generative learning, which could be applied to generative learning tasks with high performance and low computational costs.

Acknowledgements

This work is supported in part by Natural Science Foundation of Jiangxi Province of China (20232BAB212025); High-level and Urgently Needed Overseas Talent Programs of Jiangxi Province (20232BCJ25024, 20232BCJ25026); National Natural Science Foundation of China (62102133, 81960325); Kaifeng Major Science and Technology Project (21ZD011); Ji'An Finance and Science Fundation (20211085454, 20222151746, 20222151704).

References

Bao, F.; Li, C.; Zhu, J.; and Zhang, B. 2022. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. *arXiv preprint arXiv:2201.06503*.

Bińkowski, M.; Sutherland, D.; Arbel, M.; Gretton, A.; and Demystifying, M. 2018. GANs. In *International Conference on Learning Representations (ICLR)*.

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

Coates, A.; Ng, A.; and Lee, H. 2011. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 215–223. JMLR Workshop and Conference Proceedings.

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.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Jiang, Y.; Chang, S.; and Wang, Z. 2021. TransGAN: Two Pure Transformers Can Make One Strong GAN, and That Can Scale Up. *arXiv:2102.07074*.

Jolicoeur-Martineau, A.; Li, K.; Piché-Taillefer, R.; Kachman, T.; and Mitliagkas, I. 2021. Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080*.

Jolicoeur-Martineau, A.; Piché-Taillefer, R.; des Combes, R. T.; and Mitliagkas, I. 2020. Adversarial score matching and improved sampling for image generation. *arXiv:2009.05475*.

Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

Karras, T.; Aittala, M.; Aila, T.; and Laine, S. 2022. Elucidating the Design Space of Diffusion-Based Generative Models. *arXiv:2206.00364*.

Karras, T.; Laine, S.; and 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*, 4401–4410.

Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2020. Analyzing and Improving the Image Quality of StyleGAN. In *Proc. CVPR*.

Kingma, D. P.; Salimans, T.; Poole, B.; and Ho, J. 2023. Variational Diffusion Models. *arXiv:2107.00630*.

Kodali, N.; Abernethy, J.; Hays, J.; and Kira, Z. 2017. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*.

Kong, Z.; and Ping, W. 2021. On Fast Sampling of Diffusion Probabilistic Models. *arXiv:2106.00132*.

Krizhevsky, A.; et al. 2009. Learning multiple layers of features from tiny images.

Kynkänniemi, T.; Karras, T.; Laine, S.; Lehtinen, J.; and Aila, T. 2019. Improved Precision and Recall Metric for Assessing Generative Models. *arXiv:1904.06991*.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Lu, C.; Zhou, Y.; Bao, F.; Chen, J.; Li, C.; and Zhu, J. 2022. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35: 5775–5787.

Luhman, E.; and Luhman, T. 2021. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*.

Meng, C.; He, Y.; Song, Y.; Song, J.; Wu, J.; Zhu, J.-Y.; and Ermon, S. 2022. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. *arXiv:2108.01073*.

Meng, X.; and Kabashima, Y. 2023. Diffusion Model Based Posterior Sampling for Noisy Linear Inverse Problems. *arXiv:2211.12343*.

Pang, M.; Wang, B.; Cheung, Y.-m.; Chen, Y.; and Wen, B. 2021a. VD-GAN: A unified framework for joint prototype and representation learning from contaminated single sample per person. *IEEE Transactions on Information Forensics and Security*, 16: 2246–2259.

Pang, M.; Wang, B.; Ye, M.; Cheung, Y.-m.; Chen, Y.; and Wen, B. 2021b. DisP+ V: A Unified Framework for Disentangling Prototype and Variation From Single Sample per Person. *IEEE Transactions on Neural Networks and Learning Systems*.

Pei, H.; Wei, B.; Chang, K. C.-C.; Lei, Y.; and Yang, B. 2020. Geom-GCN: Geometric Graph Convolutional Networks. In *International Conference on Learning Representations*.

Pei, H.; Yang, B.; Liu, J.; and Chang, K. C.-C. 2022. Active Surveillance via Group Sparse Bayesian Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3): 1133–1148.

Radford, A.; Metz, L.; and Chintala, S. 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv:1511.06434*.

Sajjadi, M. S. M.; Bachem, O.; Lucic, M.; Bousquet, O.; and Gelly, S. 2018. Assessing Generative Models via Precision and Recall. *arXiv:1806.00035*.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training gans. *Advances in neural information processing systems*, 29.

San-Roman, R.; Nachmani, E.; and Wolf, L. 2021. Noise Estimation for Generative Diffusion Models. *arXiv:2104.02600*.

Sauer, A.; Chitta, K.; Müller, J.; and Geiger, A. 2021. Projected gans converge faster. *Advances in Neural Information Processing Systems*, 34: 17480–17492.

Sauer, A.; Schwarz, K.; and Geiger, A. 2022. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, 1–10.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Song, J.; Vahdat, A.; Mardani, M.; and Kautz, J. 2022a. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*.

Song, Y.; Dhariwal, P.; Chen, M.; and Sutskever, I. 2023. Consistency models.

Song, Y.; Durkan, C.; Murray, I.; and Ermon, S. 2021. Maximum Likelihood Training of Score-Based Diffusion Models. *arXiv:2101.09258*.

Song, Y.; and Ermon, S. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32.

Song, Y.; Shen, L.; Xing, L.; and Ermon, S. 2022b. Solving Inverse Problems in Medical Imaging with Score-Based Generative Models. *arXiv:2111.08005*.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Wang, Z.; Zheng, H.; He, P.; Chen, W.; and Zhou, M. 2022. Diffusion-GAN: Training GANs with Diffusion. *arXiv preprint arXiv:2206.02262*.

Wu, Y.; Donahue, J.; Balduzzi, D.; Simonyan, K.; and Lillicrap, T. 2019. Logan: Latent optimisation for generative adversarial networks. *arXiv preprint arXiv:1912.00953*.

Yang, L.; Zhang, Z.; Song, Y.; Hong, S.; Xu, R.; Zhao, Y.; Zhang, W.; Cui, B.; and Yang, M.-H. 2023. Diffusion Models: A Comprehensive Survey of Methods and Applications. *arXiv:2209.00796*.

Zheng, H.; Nie, W.; Vahdat, A.; Azizzadenesheli, K.; and Anandkumar, A. 2023. Fast Sampling of Diffusion Models via Operator Learning. *arXiv:2211.13449*.