SPI-GAN: DENOISING DIFFUSION GANS WITH STRAIGHT-PATH INTERPOLATIONS

Jinsung Jeon
Yonsei University
jjsjjs0902@yonsei.ac.kr

Noseong Park
KAIST
noseong@kaist.ac.kr

ABSTRACT

Score-based generative models (SGMs) show the state-of-the-art sampling quality and diversity. However, their training/sampling complexity is notoriously high due to the highly complicated forward/reverse processes, so they are not suitable for resource-limited settings. To solving this problem, learning a simpler process is gathering much attention currently. We present an enhanced GAN-based denoising method, called SPI-GAN, using our proposed *straight-path interpolation* definition. To this end, we propose a GAN architecture i) denoising through the straight-path and ii) characterized by a continuous mapping neural network for imitating the denoising path. This approach drastically reduces the sampling time while achieving as high sampling quality and diversity as SGMs. As a result, SPI-GAN is one of the best-balanced models among the sampling quality, diversity, and time for CIFAR-10, and CelebA-HQ-256.

1 INTRODUCTION

Generative models are one of the most popular research topics for deep learning. In particular, the diffusion models (Song & Ermon, 2019; Ho et al., 2020), which progressively corrupt original data and revert the corruption process to generate an image, have recently shown good performance. Song et al. (2021b) proposed a stochastic differential equation (SDE)-based mechanism that embraces all those models and coined the term, *score-based generative models* (SGMs).

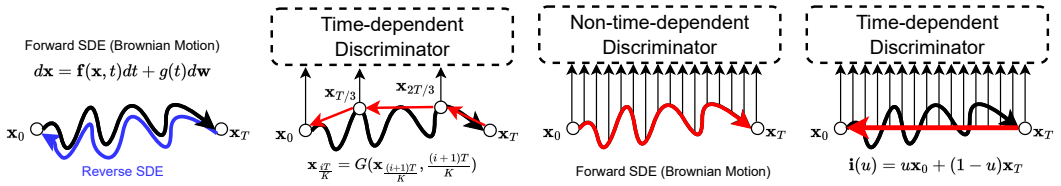
Although SGMs show good performance in terms of sampling quality and sampling diversity, they take a lot of time for sampling, which is one of the generation task trilemmas. This problem becomes more problematic in resource-limited environments, and for this reason, reducing the complexity of SGM during sampling is gathering much attention. There exist two different directions for this: i) learning a simpler process than the complicated forward/reverse process of SGMs (Das et al., 2023), and ii) letting GANs imitate [1] SGMs (Xiao et al., 2021; Wang et al., 2022).

Our method can be considered a hybrid of the previous works. Inspired by them, we propose a GAN-base hybrid method that approximates the linear interpolation. Our method imitates a denoising process following the straight-path interpolation guided by $u\mathbf{x}_0 + (1-u)\mathbf{x}_T$, where $0 \leq u \leq 1$ (cf. Figure 1 (d)). Therefore, we call our method *straight-path interpolation GAN* (SPI-GAN).

One may consider that SPI-GAN is similar to Denoising Diffusion GAN (DD-GAN) and Diffusion-GAN (cf. Figure 1 (b-d)). DD-GAN (Xiao et al., 2021) effectively reduces the number of denoising steps by letting its GAN approximate the shortcuts. In Diffusion-GAN (Wang et al., 2022), image augmentation is performed by injecting noisy images following the forward path of SGMs, and the standard adversarial training is conducted for its GAN. However, our SPI-GAN is technically different and more sophisticated in the following points:

- Our straight-path interpolation is as simple as Eq. 2, is much easier to learn. It is also possible to derive its ordinary differential equation (ODE)-based formulation, which we call the straight-path interpolation process.

[1] Let $\mathbf{x}_0$ be a clean original sample and $\mathbf{x}_T$ be a noisy sample that follows a Gaussian prior under the context of SGMs. These imitation methods learn a denoising process following the reverse SDE path by training their conditional generators to read $\mathbf{x}_t$ and output $\mathbf{x}_{t-j}$. Typically, a large $j > 0$ is preferred to reduce the denoising step (see Section 2 for a detailed explanation).



(a) The SDE-based workflow of SGMs. The reverse SDE is a generation process.

(b) Approximating the reverse SDE with K shortcuts by DD-GAN, e.g., $K = 3$ in this example.

(c) Augmentation following the forward SDE of Diffusion-GAN.

(d) Our proposed denoising method, SPI-GAN, with straight-path interpolations.

Figure 1: The comparison among four models: i) the original formulation of SGMs in (a), ii) DD-GAN's learning the shortcuts of the reverse SDE in (b), iii) Diffusion-GAN's augmentation method in (c) and iv) SPI-GAN's learning the straight path in (d). The red paths in (b), (c), and (d) are used as the discriminators' input.

- In DD-GAN, the generator learns K shortcuts through the reverse path of SGMs, whereas SPI-GAN learns the straight-interpolation path.
- Diffusion-GAN's discriminator is trained with the augmented noisy images without being conditioned on time — in other words, it is non-time-dependent. SPI-GAN's time-dependent discriminator learns the straight-path information, being conditioned on time.
- In order to learn a simple process, i.e., our straight-path interpolation, SPI-GAN uses a special neural network architecture, characterized by a mapping network.
- After all these efforts, SPI-GAN is a GAN-based method that imitates the straight-path interpolation. However, our mapping network is designed for this purpose, allowing SPI-GAN to generate fake images directly without recursion.

DD-GAN, Diffusion-GAN, and SPI-GAN attempt to address the trilemma of the generative model using GANs. Among those models, our proposed SPI-GAN, which learns a straight-path interpolation, shows the best balance in terms of the overall sampling quality, diversity, and time in two different resolution benchmark datasets: CIFAR-10, and CelebA-HQ-256

2 Related Work and Preliminaries

Neural ordinary differential equations (NODEs). Neural ordinary differential equations (Chen et al., 2018) use the following equation to define the continuous evolving process of the hidden vector $\mathbf{h}(u)$:

$$\mathbf{h}(u) = \mathbf{h}(0) + \int_0^u f(\mathbf{h}(t), t; \boldsymbol{\theta}_{\mathbf{f}})dt, \tag{1}$$

where the neural network $f(\mathbf{h}(t), t; \boldsymbol{\theta}_{\mathbf{f}})$ learns $\frac{d\mathbf{h}(t)}{dt}$. To solve the integral problem, we typically rely on various ODE solvers. The explicit Euler method is one of the simplest ODE solvers. The 4th order Runge–Kutta (RK4) method is a more sophisticated ODE solver and the Dormand–Prince (Dormand & Prince, 1980) method is an advanced adaptive step-size solver. The NODE-based continuous-time models (Chen et al., 2018; Kidger et al., 2020) show good performance in processing sequential data.

Letting GANs imitate SGMs. Among the ways to solve the SGMs trilemma, letting GANs imitate SGMs shows the best-balanced performance in terms of the generative task trilemma. Figure 1 (b) shows the key idea of DD-GAN, which proposed to approximate the reverse SDE process with K shortcuts. They internally utilize a GAN-based framework conditioned on time (step) t to learn the shortcuts. A generator of DD-GAN receives noise images as input and denoises it. In other words, the generator generates clean images through denoising K times from the noise images. To this end, DD-GAN uses a conditional generator, and $\mathbf{x}_t$ is used as a condition to generate $\mathbf{x}_{t-1}$. It repeats this K times to finally creates $\mathbf{x}_0$. For its adversarial learning, the generator can match $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)$ and $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$. Figure 1 (c) shows the overall method of Diffusion-GAN. In contrast to DD-GAN, Diffusion-GAN directly generates clean images and its discriminator is trained after augmenting real images with noisy images that follow the forward SDE from the generated images. The noisy images are used as input to the discriminator to prevent mode-collapse in GANs.

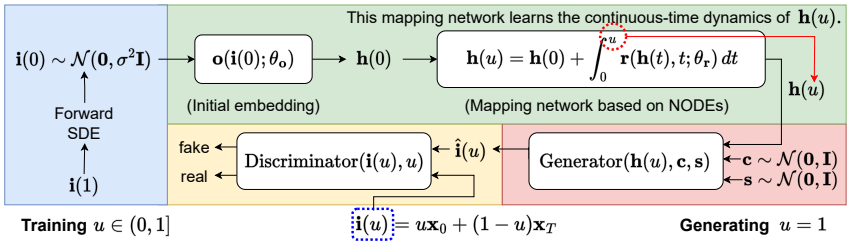


Figure 2: The architecture of our proposed SPI-GAN. $\mathbf{h}(u)$ is a latent vector which generates an interpolated image $\mathbf{i}(u)$ at time u. Therefore, $\mathbf{i}(1)$ is an original image and $\mathbf{i}(0)$ is a noisy image. We perform this adversarial training every time u but generate images with $u = 1$. The constant noise $\mathbf{c}$ and the layer-wise varying noise $\mathbf{s}$ enable the stochasticity of the generator.

3 Proposed Method

Before describing our proposed method, the notations in this paper are defined as follows: i) $\mathbf{i}(u) \in \mathbb{R}^{C\times H\times W}$ is an image with a channel C, a height H, and a width W at interpolation point u. ii) $\hat{\mathbf{i}}(u) \in \mathbb{R}^{C\times H\times W}$ is a generated fake image at interpolation point u. iii) $\mathbf{h}(u) \in \mathbb{R}^{\dim(\mathbf{h})}$ is a latent vector at interpolation point u. iv) $\mathbf{r}(\mathbf{h}(t), t; \boldsymbol{\theta}_\mathbf{r}) \in \mathbb{R}^{\dim(\mathbf{h})}$ is a neural network approximating the time derivative of $\mathbf{h}(t)$, denoted $\frac{d\mathbf{h}(t)}{dt}$. Our proposed SPI-GAN consists of four parts, each of which has a different color in Figure 2

3.1 Diffusion through the forward SDE

In the first part highlighted in blue, σ^2 can be different for $\mathbf{i}(0) \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ depending on the type of the forward SDE. Unlike the reverse SDE, which requires step-by-step computation, the forward SDE can be calculated with one-time computation for a target time t (Song et al., 2021b). Therefore, it takes $\mathcal{O}(1)$ in the first part. We note that $\mathbf{i}(0) = \mathbf{x}_T$ and $\mathbf{i}(1) = \mathbf{x}_0$ are the two ends of the straight-path interpolation which will be described shortly.

3.2 Straight-path interpolation

SPI-GAN has advantages over SGMs since it learns a much simpler process than the SDE-based path of SGMs. Our straight-path interpolation (SPI) is defined as follows:

$$\mathbf{i}(u) = u\mathbf{x}_0 + (1-u)\mathbf{x}_T, \tag{2}$$

where $u \in (0, 1]$ is an intermediate point and therefore, $\mathbf{i}(1) = \mathbf{x}_0$ and $\mathbf{i}(0) = \mathbf{x}_T$. We use a straight-path between $\mathbf{i}(1)$ and $\mathbf{i}(0)$ to learn the shortest path with the minimum Wasserstein distance. Therefore, Our straight-path interpolation process is simpler and thus more suitable for neural networks to learn. During our training process, we randomly sample u every iteration.

3.3 Mapping network

The mapping network, which generates a latent vector $\mathbf{h}(u)$, is the most important component in our model. The mapping network consists of an initial embedding network, denoted $\mathbf{o}$, that generates the initial hidden representation $\mathbf{h}(0)$ from $\mathbf{i}(0)$, and a NODE-based mapping network. The role of network $\mathbf{o}$ is to reduce the size of the input to the mapping network for decreasing sampling time. In addition, the NODE-based mapping network generates the latent vector for a target interpolation point u, whose initial value problem (IVP) is defined as follows:

$$\mathbf{h}(u) = \mathbf{h}(0) + \int_0^u \mathbf{r}(\mathbf{h}(t), t; \boldsymbol{\theta}_\mathbf{r})dt, \tag{3}$$

where $\frac{d\mathbf{h}(t)}{dt} = \mathbf{r}(\mathbf{h}(t), t; \boldsymbol{\theta}_\mathbf{r})$, and $\mathbf{r}$ has multiple fully-connected layers in our implementation. In general, $\mathbf{h}(0)$ is a lower-dimensional representation of the input. One more important point is that we maintain a single latent space for all u and therefore, $\mathbf{h}(u)$ has the information of the image to generate at a target interpolation time u. For instance, Figure 3 shows that a noisy image is generated from $\mathbf{h}(0)$ but a clean image from $\mathbf{h}(1)$.

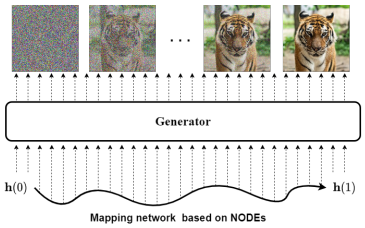


Figure 3: Process of generating sample from continuous latent vector.

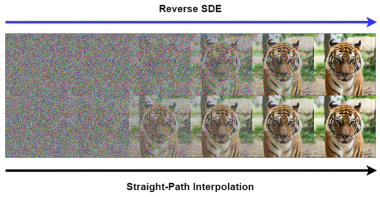


Figure 4: Difference between reverse SDE and interpolation.

3.4 GENERATOR

We customize the generator architecture of StyleGAN2 (Karras et al., 2020b) for our purposes. Our generator takes $\mathbf{c}, \mathbf{s}$ as input and mimics the stochastic property of SGMs. However, the biggest difference from StyleGAN2 is that our generator is trained by the *continuous-time* mapping network which generates latent vectors at various interpolation points while maintaining one latent space across them. This is the key point in our model design to generate the interpolated image $\hat{\mathbf{i}}(u)$ with various u settings.

3.5 DISCRIMINATOR

The discriminator of SPI-GAN is time-dependent, unlike the discriminator of traditional GANs. It takes $\hat{\mathbf{i}}(u)$ and the embedding of u as input and learns to classify images from various interpolation points. As a result, it i) learns the data distribution which changes following the straight-path interpolation, and ii) solves the mode-collapse problem that traditional GANs have, since our discriminator sees various clean and noisy images. DD-GAN and Diffusion-GAN also use this strategy to overcome the mode-collapse problem. They give noise images following the reverse (forward) SDE path as input to the discriminator. As shown in Figure 4, however, the straight-path interpolation method maintains a better balance between noisy and clean images than the case where we sample following the SDE path. Therefore, our discriminator learns a more balanced set of noisy and clean images than that of DD-GAN and Diffusion-GAN.

3.6 HOW TO GENERATE

In order to generate samples with SPI-GAN, we need only $\mathbf{h}(1)$ from the mapping network. Unlike other auto-regressive denoising models that require multiple steps when generating samples, e.g., DD-GAN (cf. Figure 1 (b)), SPI-GAN learns the denoising path using a NODE-based mapping network. After sampling $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$, we feed them into the mapping network to derive $\mathbf{h}(1)$ — we solve the initial value problem in the Eq. 3 from 0 to 1 — and our generator generates a fake image $\hat{\mathbf{i}}(1)$. In other words, it is possible to generate latent vector $\mathbf{h}(1)$ directly in our case, which is later used to generate a fake sample $\hat{\mathbf{i}}(1)$.

4 EXPERIMENTS

For evaluating our proposed method, SPI-GAN, we use different resolution benchmark datasets. CIFAR-10 (Krizhevsky et al., 2014) has a resolution of 32x32. CelebA-HQ-256 (Karras et al., 2018) contains high-resolution images of 256x256. We use 5 evaluation metrics to quantitatively evaluate fake images. The inception score (Salimans et al., 2016) and the Fréchet inception distance (Heusel et al., 2017) are traditional methods to evaluate the fidelity of fake samples. The improved recall (Kynkäänniemi et al., 2019) reflects whether the variation of generated data matches the variation of training data. Finally, the number of function evaluations (NFE).

4.1 MAIN RESULTS

Table 1: Results of the generation on CIFAR-10.

Model	IS ↑	FID ↓	Recall↑	NFE↓
SPI-GAN (ours)	**10.2**	3.01	**0.66**	1
Diffusion-GAN (StyleGAN2) (Wang et al., 2022)	9.94	3.19	0.58	1
Denoising Diffusion GAN (DD-GAN), $K = 4$ (Xiao et al., 2021)	9.63	3.75	0.57	4
Score SDE (VP) (Song et al., 2021b)	9.68	2.41	0.59	2000
DDPM (Ho et al., 2020)	9.46	3.21	0.57	1000
NCSN (Song & Ermon, 2019)	8.87	25.3	-	1000
Score SDE (VE) (Song et al., 2021b)	9.89	2.20	0.59	2000
LSGM (Vahdat et al., 2021)	9.87	**2.10**	0.61	147
DDIM, T=50 (Song et al., 2021a)	8.78	4.67	0.53	50
FastDDPM, T=50 (Kong & Ping, 2021)	8.98	3.41	0.56	50
Recovery EBM (Gao et al., 2021)	8.30	9.58	-	180
Improved DDPM (Nichol & Dhariwal, 2021)	-	2.90	-	4000
StyleGAN2 w/o ADA (Karras et al., 2020b)	9.18	8.32	0.41	1
StyleGAN2 w/ ADA (Karras et al., 2020a)	9.83	2.92	0.49	1
StyleGAN2 w/ Diffaug (Zhao et al., 2020)	9.40	5.79	0.42	1

Table 2: Results on CelebA-HQ-256.

Model	FID↓
SPI-GAN (ours)	**6.62**
DD-GAN	7.64
Score SDE	7.23
LSGM	7.22
UDM	7.16
PGGAN (Karras et al., 2018)	8.03
Adv. LA (Pidhorskyi et al., 2020)	19.2
VQ-GAN (Esser et al., 2021)	10.2
DC-AE (Parmar et al., 2021)	15.8

In this subsection, we evaluate our proposed model quantitatively and qualitatively. For CIFAR-10, we perform the unconditional image generation task for fair comparisons with existing models. The quantitative evaluation results are shown in Table 1. Although our Fréchet inception distance (FID) is 0.6 worse than that of the Score SDE (VP), it shows better scores in all other metrics. However, our method has a better FID score than that of DD-GAN and Diffusion-GAN, which are the most related methods. Diffusion-GAN, which follows the reverse SDE path using GAN, is inferior to our method for all those three quality metrics. LGSM also shows high quality for FID. However, its inception score (IS) and improved recall (Recall) scores are worse than ours.

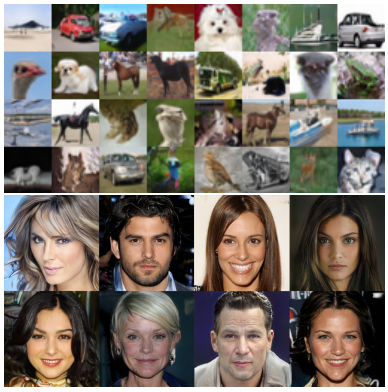


Figure 5: Qualitative results on CIFAR-10, and CelebA-HQ-255.

Even for high-resolution images, our model shows good performance. In particular, our method shows the best FID score for CelebA-HQ-256 in Table 2, which shows the efficacy of our proposed method. As shown, our method is able to generate visually high-quality images.

4.2 Additional Studies

Sampling time analyses. Our model shows outstanding performance in all evaluation metrics compared to DD-GAN and Diffusion-GAN. In particular, SPI-GAN, unlike DD-GAN, does not increase the sample generation time — in fact, our method only affects the training time because we train our method with $u \in (0, 1]$. However, we always use $u = 1$ for generating a clean image $\hat{\mathbf{i}}(1)$. Therefore, our method is fast in generating images after being trained, which is one good characteristic of our method. We measure the wall-clock runtime 10 times for CIFAR-10 with a batch size of 100 using an A5000 GPU to evaluate the sampling time. We applied the same environment for fair comparisons. As a result, our method's sample generation time in Table 3 is almost the same as that of StyleGAN2, which is one of the fastest methods. In summary, SPI-GAN not only increases the quality of samples but also decreases the sampling time.

Table 3: The Generation time comparison

Model	Time
SPI-GAN (Ours)	0.04
Diffusion-GAN (StyleGAN2)	0.04
DD-GAN	0.36
StyleGAN2	0.04

5 Conclusions and Discussions

Score-based generative models (SGMs) now show the state-of-the-art performance in image generation. However, the sampling time of SGMs is significantly longer than other generative models, such as GANs, VAEs, and so on. Therefore, we presented the most balanced model by reducing the sampling time, called SPI-GAN. Our method is a GAN-based approach that imitates the straight-path interpolation. Moreover, it can directly generate fake samples without any recursive computation (or step). Our method shows the best sampling quality in various metrics and faster sampling time than other score-based methods. All in all, one can see that SPI-GAN is one of the most balanced methods among the generative task trilemma's criteria: sampling quality, diversity, and time.

REFERENCES

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Ayan Das, Stathi Fotiadis, Anil Batra, Farhang Nabiei, FengTing Liao, Sattar Vakili, Da-shan Shiu, and Alberto Bernacchia. Image generation with shortest path diffusion. *arXiv preprint arXiv:2306.00501*, 2023.

J.R. Dormand and P.J. Prince. A family of embedded runge-kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19 – 26, 1980.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12873–12883, 2021.

Ruiqi Gao, Yang Song, Ben Poole, Ying Nian Wu, and Diederik P Kingma. Learning energy-based models by diffusion recovery likelihood. *arXiv preprint arXiv:2012.08125*, 2021.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2018.

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

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8110–8119, 2020b.

Patrick Kidger, James Morrill, James Foster, and Terry J. Lyons. Neural controlled differential equations for irregular time series. In *NeurIPS*, 2020.

Zhifeng Kong and Wei Ping. On fast sampling of diffusion probabilistic models. *arXiv preprint arXiv:2106.00132*, 2021.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 55(5), 2014.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

Gaurav Parmar, Dacheng Li, Kwonjoon Lee, and Zhuowen Tu. Dual contradistinctive generative autoencoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 823–832, 2021.

Stanislav Pidhorskyi, Donald A Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14104–14113, 2020.

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

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2021a.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2021b.

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34, 2021.

Zhendong Wang, Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Diffusion-gan: Training gans with diffusion. *arXiv preprint arXiv:2206.02262*, 2022.

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans. *arXiv preprint arXiv:2112.07804*, 2021.

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