Exploring Guided Sampling of Conditional GANs

Yifei Zhang[1†], Mengfei Xia[2†], Yujun Shen[3], Jiapeng Zhu[4], Ceyuan Yang[5], Kecheng Zheng[3], Lianghua Huang[6], Yu Liu[6], and Fan Cheng[1‡]

[1]Shanghai Jiao Tong Univeristy [2]Tsinghua University [3]Ant Group [4]HKUST [5]Shanghai AI Laboratory [6]Alibaba Group

Abstract. Guided sampling serves as a widely used inference technique in diffusion models to trade off sample fidelity and diversity. In this work, we confirm that generative adversarial networks (GANs) can also benefit from guided sampling, not even requiring to pre-prepare a classifier (*i.e.*, classifier guidance) or learn an unconditional counterpart (*i.e.*, classifier-free guidance) as in diffusion models. Inspired by the organized latent space in GANs, we manage to estimate the data-condition joint distribution from a well-learned conditional generator simply through vector arithmetic. With such an *easy implementation*, our approach, termed `GANdance`, improves the FID score of a state-of-the-art GAN model pre-trained on ImageNet 64×64 from 8.87 to 6.06, barely increasing the inference time. We then propose a learning-based variant of our framework to better approximate the distribution of the entire dataset, further improving the FID score to 4.37. It is noteworthy that our sampling strategy sufficiently closes the gap between GANs and one-step diffusion models (*i.e.*, with FID 4.02) under comparable model size. Code is available at https://github.com/zyf0619sjtu/GANdance.

Keywords: Generative adversarial networks · Conditional generation · Guided sampling

1 Introduction

Generative models have enabled a wide range of real-world applications in the past few years, such as stimulating creativity [20, 36, 38, 64], editing visual assets [40], and entertainments [17]. Among all types of generative paradigms, like variational autoencoders [27], autoregressive models [34, 56], and normalizing flows [37], diffusion models [51] tend to become the dominate solution and are much sought after by the community. Thanks to better mode coverage and stronger scalability, diffusion models even beat the previous state-of-the-art generative framework, *i.e.*, generative adversarial networks (GANs), especially when the data is with a broad distribution [8].

† Equal contribution.
‡ Corresponding author.

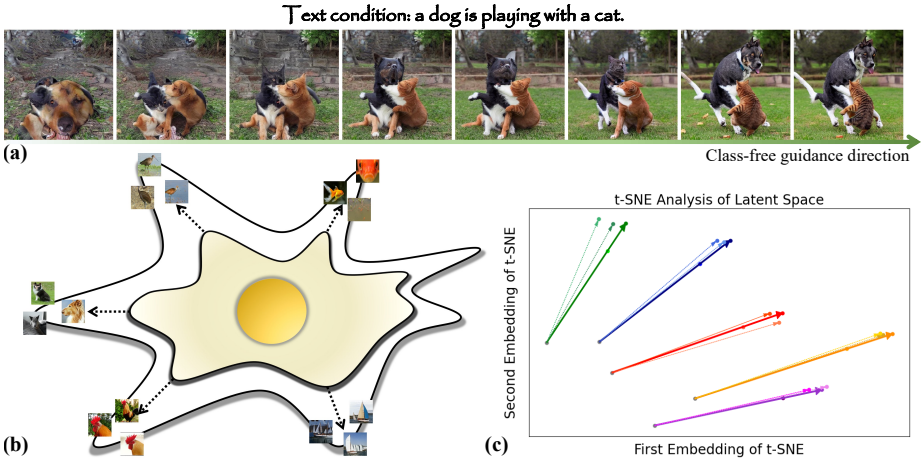

Fig. 1: (a) **Visualization** of guided sampling through Stable Diffusion 1.5 [39], in which the state-of-the-art DPM fails without guidance. (b) **Motivation scheme** of `GANdance`. The yolk represents the joint data distribution without condition effect, while the egg white stands for the conditional distribution. The guiding direction from joint to conditional distribution will strengthen the condition fidelity. (c) **Visualization** of the latent space of GANs via t-SNE decomposition. Each collection of similar colors represents a single class, while gray dots represent the joint distribution. All bold solid lines stand for the guiding direction from joint to conditional distribution, while all other dotted lines indicate the guiding directions achieved by `GANdance`.

However, we notice that the success of diffusion models stems from a post-training technique, *i.e.*, guided sampling [8, 15], to some extent. As we can see in Fig. 1a, Stable Diffusion 1.5 [39] fails to produce meaningful images without any guidance. The key idea of guided sampling is to rectify the predicted noise with the gradient of a pre-prepared classifier, which is widely known as classifier guidance [8]. Intuitively, thanks to such an inference technique, the model is relaxed from "exactly" reproducing the data distribution, as the outside curve shown in Fig. 1b. Instead, it only needs to learn a direction from the unconditional generation (the center circle) to the conditional generation (the middle curve) such that moving along this direction will finally decode the condition satisfyingly (the dashed arrows). Ho and Salimans [15] further proposed classifier-free guidance to circumvent the reliance on the classifier, yet requiring a jointly learned unconditional model to estimate the moving direction.

In this work, we would like to figure out whether GANs can also benefit from a better sampling strategy to match the performance of diffusion models. Encouragingly, our answer is a *big yes*, at least under the data scale at the ImageNet [7] level. Our motivation is that the latent space of a GAN is usually well-organized and hence allows easy semantic editing via vector arithmetic [46, 48, 62]. Inspired by this, we propose to estimate the unconditional generation from a conditional generator by eradicating the effect of the conditions. Concretely, given a sampled

noise, we first fuse it with all conditions to obtain a collection of conditional latents, then average these latents to cancel the conditional effect, and finally use the latent with respect to the target class and the averaged result to compute the guiding direction. We call the above process `GANdance` as it offers *GAN* sampling gui*dance*. We further propose a training-based variant of our framework, which asks the generator to better approximate the distribution of the entire dataset via learning an additional class. That way, at the inference stage, we can directly compute the guiding direction by fusing the sampled noise with the target class and the newly introduced class.

We evaluate our approach on a range of state-of-the-art GANs, including StyleGAN2 [1], BigGAN [4]and the large-scale Aurora [64]. On the ImageNet 64×64 dataset, our training-free approach is capable of improving the FID score [13] of Aurora from 8.87 to 6.06. It is noteworthy that, unlike classifier-free guidance of diffusion models that doubles the inference time, `GANdance` barely affects the sampling speed. We also analyze the sampling process and confirm that our computed direction indeed pushes the model towards the conditional distribution, as shown in Fig. 1c. With the proposed training-based framework, we manage to further boost the performance from 6.06 to 4.37, almost on par with the state-of-the-art one-step diffusion model (*i.e.*, with FID 4.02) under the same model size. We hope that our discovery could bring GANs back to the public view and encourage more studies in the field of visual content generation.

2 Related Work

Generative adversarial networks. Formulated as a two-player game between a generator and a discriminator, GAN [9] is designed to model a mapping from a known distribution to observed data distribution through adversarial training. Thanks to the sophisticated model [24] and training design [2, 21], GANs have demonstrated excellent performance in various visual generation tasks, such as image generation [4, 21, 23–25], video generation [41, 55], and 3D-aware image synthesis [5, 6, 11, 33, 45, 49, 58]. In particular, style-based GANs [23–25] have shown impressive ability on single-domain high-resolution images and interpretable latent space [46, 63]. In addition, some studies focus on exploring the use of GAN for conditional generation [32], including signals like class labels [4, 44, 61], texts [20, 43, 54, 64], and reference images [19, 29, 60]. Although GAN has succeeded in many of the above fields, its performance is still unsatisfactory when facing diverse conditional generation tasks.

Diffusion models and guided sampling. As an emerging type of generative model, DPMs [14,50,51] has achieved remarkable results in many fields including image generation [18], image editing [40] and video synthesis [16]. In recent years, DPMs have outperformed Generative Adversarial Networks (GANs) by a considerable margin [8], especially in the open-vocabulary text-to-image domain [3, 36]. The success of DPMs in high-quality conditional image generation can be attributed to two main factors: LDM [39] compresses high-resolution images into a lower-dimensional space to reduce the optimization difficulty of

DPM: on the other hand, effective guided sampling methods [8, 15] have further enhanced the quality of synthesis images. Classifier guidance [8] utilizes a pre-trained classifier to steer the diffusion process using gradients. Classifier-free guidance [15], building on this, removes the reliance on the pre-trained classifier by jointly training the model on both conditional and unconditional generation tasks, allowing for broader application in open-vocabulary conditional generation tasks. In contrast to these guidance methods used in DPMs, our method leverages the inherent nature of GANs, which possess a well-organized latent space, without depending on classifiers or additional training. With a very easy implementation, we can guide GANs to achieve even better results.

3 Method

3.1 Revisiting GANs and Conditional GANs.

GANs were first proposed by Goodfellow [10], by involving a generator G and a discriminator D. Formally, for the vanilla GAN, let $\mathbf{x}$ be the training data with an unknown distribution $q(\mathbf{x})$, GANs are devoted to mapping a random noise $\mathbf{z}$ to sample using G, while discriminating real or generated samples through D, respectively. The GAN training endeavors to reach Nash equilibrium via the following two losses:

$$\mathcal{L}_G = -\mathbb{E}_{\mathbf{z}}[\log D(G(\mathbf{z}))], \tag{1}$$

$$\mathcal{L}_D = -\mathbb{E}_{\mathbf{x}}[\log D(\mathbf{x})] - \mathbb{E}_{\mathbf{z}}[\log(1 - D(G(\mathbf{z})))], \tag{2}$$

where $\mathbf{z}$ is random noise embedded in the latent space.

Follow-up seminal works [23, 24, 26] introduced the *style space* to GANs to achieve further improvement on the sampling quality, enabling GANs to be a prominent paradigm of generative model. Concretely, style-based GANs divide the generator G into two parts, *i.e.*, G_{map} and G_{syn}. A randomly sampled latent code $\mathbf{z}$ will be first mapped to a style code $\mathbf{w}$ in the disentangled latent space, *i.e.*, $\mathcal{W}$ space. Then G_{syn} injects $\mathbf{w}$ into each layer, outputting the synthesized sample. Thanks to this design, the highly depressed $\mathcal{W}$ space is confirmed to be well-organized with a hierarchical structure, enjoying great interpretability [46,57,62].

Despite the expeditious generation on single-domain datasets (*e.g.*, human faces), conditional generation utilizing GANs remains not well-explored. Conditional GANs are designed to approximate the marginal distribution given the condition c by injecting the condition information into both generator and discriminator [32]. By doing so, we can rewrite the GAN losses as below:

$$\mathcal{L}_G = -\mathbb{E}_{\mathbf{z},c}[\log D(G(\mathbf{z}, c), c)], \tag{3}$$

$$\mathcal{L}_D = -\mathbb{E}_{\mathbf{x},c}[\log D(\mathbf{x}, c)] - \mathbb{E}_{\mathbf{z},c}[\log(1 - D(G(\mathbf{z}, c), c))]. \tag{4}$$

However, existing conditional GANs are usually criticized for unsatisfactory visual quality and limited diversity, especially compared with DPMs such as ADM [8]. In the sequel, we will focus on the conditional generation utilizing GANs, arguing that GANs are sufficiently capable of the task thanks to the well-organized latent space.

3.2 Training-Free Guidance in Latent Space

Before stating the guidance strategy, we first review the classifier guidance based on Bayesian theory. Given fixed condition c, we have the following relationship on the conditional probability $p(\mathbf{x}|c)$ according to Bayesian theory:

$$\log p(\mathbf{x}|c) = \log p(\mathbf{x}) + \log p(c|\mathbf{x}) - \log p(c), \tag{5}$$

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}|c) = \nabla_{\mathbf{x}} \log p(\mathbf{x}) + \nabla_{\mathbf{x}} \log p(c|\mathbf{x}), \tag{6}$$

in which $p(c|\mathbf{x})$ represents the probability of $\mathbf{x}$ classified to be with c. Inspired by Eq. (6), the conditional generation can be improved when increasing $p(c|\mathbf{x})$ by adding classifier gradient $\nabla_{\mathbf{x}} \log p(c|\mathbf{x})$ multiplied by a scale λ.

Recall that in GAN editing, thanks to the well-organized latent space with continuous semantics, one can strengthen a given attribute for an image by simply moving the latent code linearly with the corresponding direction. Theoretically, denote by c the given attribute, there exists an attached direction in the latent space denoted by $\mathbf{n}$, which dramatically increases $p(c|\mathbf{x})$ for *any* $\mathbf{x}$ by linear vector interpolation with an editing strength λ. This implies the close relationship between the latent space and the probability in the image space.

Formally, we have the following theorem by some assumptions:

Theorem 1. *Assume that $G_{syn} : \mathcal{W} \to G_{syn}(\mathcal{W})$ is bijective, it induces the probability from the image space $G_{syn}(\mathcal{W})$ up to $\mathcal{W}$ by*

$$p(\mathbf{w}|c) = p(G_{syn}(\mathbf{w})|c) \cdot \det J(\mathbf{w}), \tag{7}$$

$$p(\mathbf{w}) = p(G_{syn}(\mathbf{w})) \cdot \det J(\mathbf{w}), \tag{8}$$

$$p(c|\mathbf{w}) = p(c|G_{syn}(\mathbf{w})), \tag{9}$$

in which $\det J(\mathbf{w})$ is the determinant of the Jacobian matrix of G_{syn} over $\mathbf{w}$. Then we have the following equality:

$$\log p(\mathbf{w}|c) = \log p(\mathbf{w}) + \log p(c|\mathbf{w}) - \log p(c), \tag{10}$$

$$\nabla_{\mathbf{w}} \log p(\mathbf{w}|c) = \nabla_{\mathbf{w}} \log p(\mathbf{w}) + \nabla_{\mathbf{w}} \log p(c|\mathbf{w}), \tag{11}$$

Theorem 1 confirms to bridge the latent space and the conditional probability in image space, and hence the feasibility of GAN editing by selecting one single direction in the latent space. Re-scoring analysis [46] verifies this theorem in practice, which calculates the difference of the classifier output between images before and after editing. In other words, the pre-selected appropriate direction $\mathbf{n}$ in the latent space resembles the gradient $\nabla_{\mathbf{w}} \log p(c|\mathbf{w})$.

Revealing this neat but insightful mathematical foundation, we first propose the training-free version of `GANdance`, which is an intuitive guided sampling strategy for conditional generation of GANs, and implemented in a plug-in and training-free fashion. To be more detailed, in order to strengthen the condition fidelity, motivated by the theory above, we point out that one simply needs to *set the joint data distribution in $\mathcal{W}$ space as the opposite of the given condition.*

Alg. 1 Training-free Guidance

Require: $\mathbf{z}, c, \lambda$: guidance strength, $\mathcal{C}$: opposite condition set
1: $\mathbf{w}_c = G_{map}(\mathbf{z}, c)$
2: $\mathbf{w}_{oppo} = \mathbb{E}_{c' \in \mathcal{C}}[G_{map}(\mathbf{z}, c')]$
3: $\mathbf{w}'_c = \mathbf{w}_{oppo} + \lambda(\mathbf{w}_c - \mathbf{w}_{oppo})$
4: $\mathbf{x}_c = G_{syn}(\mathbf{w}'_c)$
5: **return** $\mathbf{x}_c$

Alg. 2 Training-based Guidance

Require: $\mathbf{z}, c, \lambda$: guidance strength, $\varnothing$: additional condition
1: $\mathbf{w}_c = G_{map}(\mathbf{z}, c)$
2: $\mathbf{w}_{oppo} = G_{map}(\mathbf{z}, \varnothing)$
3: $\mathbf{w}'_c = \mathbf{w}_{oppo} + \lambda(\mathbf{w}_c - \mathbf{w}_{oppo})$
4: $\mathbf{x}_c = G_{syn}(\mathbf{w}'_c)$
5: **return** $\mathbf{x}_c$

Guaranteed by Theorem 1, such an operation increases $\log p(c|\mathbf{w})$ rapidly, and will significantly improve the conditional generation quality by increasing $p(\mathbf{w}|c)$.

Estimating the joint data distribution from a well-learned conditional generator can be implemented easily. Given a randomly sampled noise $\mathbf{z}$ and a pre-trained generator G, we can estimate the joint data distribution for $\mathbf{z}$ by fusing it with all conditions to reach a collection of conditional latents. Then calculating the expectation of such all conditional latents will cancel the conditional effect, which becomes an approximate of the joint data distribution. Finally, one can use the latent with respect to the target class and the averaged result to compute the guiding direction, demonstrated in Alg. 1. To make a further step, by drawing lessons from the theory of probability, we argue that the average conditional latents over even only part of all conditions also serves as a great estimation, since expectation over all potential average conditional latents over part of conditions equals the mean over all conditions. Therefore, it is feasible to first uniformly sample a subset $\mathcal{C}$ of conditions (termed as the *opposite condition set*), and then calculate expectation over $\mathcal{C}$. This decreases the time cost to traverse all conditions with almost no performance degradation. We report the quantitative results in Sec. 4.2. It is noteworthy that, the linear interpolation using the difference of $(\mathbf{w}_c - \mathbf{w}_{oppo})$ (*i.e.*, the guiding direction) not only increases $p(c|\mathbf{w})$, but also decreases $p(c'|\mathbf{w})$ for each c' in the opposite condition set $\mathcal{C}$ by interpolation via $(\mathbf{w}_c - \mathbf{w}_{oppo})$ (*i.e.*, moving away from each $G_{map}(\mathbf{z}, c')$). Benefiting from the light-weight G_{map}, our training-free guidance barely increases the inference time (especially compared with classifier or classifier-free guidance in DPMs), while significantly improving the generation quality.

3.3 Learning-Based `GANdance` for Conditional GANs

Recall that our proposed training-free guidance estimates the joint data distribution by the expectation over the opposite condition set. Based on the analysis before, more accurate estimation will lead to more effective drift direction in the latent space and facilitate conditional sampling. Therefore, we hope to achieve more accurate estimation at minimal cost. To this end, we further design a learning-based variant of our framework, dealing with the approximation of the joint distribution. Besides the condition set, we propose to leverage an

additional condition attached with the entire dataset for both the generator and the discriminator. Concretely, for each data-condition pair $(\mathbf{x}, c)$, we will reset the condition as the additional condition at a fixed probability, resembling the training methodology of classifier-guidance in DPMs [15] and supplementing the native training framework of conditional GANs.

This learning-based `GANdance` can imitate the joint distribution more accurately since it builds upon the native conditional GANs which focus on each conditional distribution via shared embedding layers in both generator and discriminator. Furthermore, with the additional condition attached to the entire dataset, the guiding direction can be obtained directly, by fusing the sampled noise with the target and the additional condition and performing subtraction, as described in Alg. 2. It is also worth noting that we only need to raise the input dimension of the embedding layers by one, with no other structural modification required. That is to say, plugging `GANdance` in GANs incurs almost no additional training cost.

On the other hand, note that the training of additional condition benefits from the shared embedding modules in both generator and discriminator. Therefore, it is possible to apply the learning-based `GANdance` on pre-trained conditional GANs by reusing all parameters and adding a tensor to the embedding module of generator and discriminator, respectively. We claim that this will further facilitate the conditional generation, since the prior of conditional distribution from pre-trained models serves as a promising warm-up for the newly added parameters. This will be addressed in *Supplementary Material*.

The additional condition involved in the native conditional GAN training might make it challenging to retain the efficacy of the original conditional generation modules. Therefore, the probability to reset the condition is attached great importance to the proposed `GANdance`. Theoretically, large probability will entail to concentrate the training more on the generation with additional condition, harming the quality of the original efficacy. On the other hand, too small probability suggests ignoring the approximation of the joint distribution, leading to inaccurate opposite and hence poor guided sampling quality. Performance comparison among different probabilities is addressed in Sec. 4.5.

3.4 Layer-Wise `GANdance`

It is well recognized that the latent space of style-based GANs [24, 59] controls the output of G layer by layer, and some channels of $\mathbf{w}$ dominate different visual attributes of the generated image [57]. In detail, StyleSpace [57] shows that the channels in $\mathbf{w}$ with respect to early layers in G often affect the high-level semantics (shape, category, etc.) of the generated image, while those channels in $\mathbf{w}$ with respect to the later layers in G often affect the low-level semantics (color, texture, etc.). A natural idea is that during the sampling process, we can apply different guidance scales to different layers in G, *i.e.*, layer-wise `GANdance`, to achieve better sampling results. To this end, we analyze the impact of guided sampling layer by layer, and the experimental results are shown in Fig. 4. We find that the part of the generator G with features below 16 x 16 resolution

can accept stronger guidance and is more likely to benefit from it. Therefore, by simply designing a layer-wise `GANdance` with decreasing guidance scales, we can further reduce the FID of an Aurora model trained on ImageNet 64 x 64 from 4.72 to **4.37**. The details will be addressed in *Supplementary Material*.

3.5 Comparison between `GANdance` and Existing Techniques

We first compare `GANdance` with the truncation trick in StyleGAN [24]. Both two tricks base on vector arithmetic, which is first proposed in DCGAN and becomes a basic operation in latent space [30]. Note that the truncation trick is designed to draw $\mathbf{w}$ away from ill-trained latent space, implemented by interpolation with expectation $\mathbf{w}$ over $\mathbf{z}$ with coefficient $\psi < 1$, enforcing each $\mathbf{w}$ to concentrate to the center of $\mathcal{W}$ space. However, `GANdance` aims to increase the posterior of generation by interpolation with unconditional latent $\mathbf{w}_{oppo}$ approximated over opposite condition set $\mathcal{C}$ with $\lambda > 1$. We argue that these two operations are compatible, in which detailed results are reported in Tab. 3.

Next, recall that the motivation of `GANdance` stands upon the Bayesian theory, aiming to improve the conditional generation by increasing $p(c|\mathbf{w})$ in $\mathcal{W}$ space. Similarly, classifier guidance in DPMs manages the same task, but equipped with an auxiliary classifier providing gradient guidance for intermediate noisy data at all timesteps [8]. Beyond supernumerary time cost for gradient calculation, existing classifier-guided DPMs struggle on the poor classification accuracy on considerably large noise strength (*e.g.*, the classifier gets an average top-1 accuracy on ImageNet 64x64 [7] less than 30% while ResNet50 [12] can easily reach 60% top-1 accuracy by finetuning from a pre-trained model).

It is also noteworthy that the sampling algorithms of `GANdance` in Algs. 1 and 2 resemble the classifier-free guidance in DPMs [15]. They introduce the similar interpolation with the difference between the conditioned latent code and its corresponding latent under joint distribution (*i.e.*, $\mathbf{w}_{oppo}$ in `GANdance` and $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \varnothing)$ in [15]). We conclude that, the difference of $(\mathbf{w}_c - \mathbf{w}_{oppo})$ is the accurate classifier gradient, while avoiding the use of a classifier thanks to surprising properties in the well-organized $\mathcal{W}$ space. On the other hand, classifier-free guidance in DPMs increases $\nabla_{\mathbf{x}} \log p(c|\mathbf{x})$ in Eq. (6) by interpolation with $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|c) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$, based on the following equality:

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|c) = -\frac{1}{\sigma_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, c), \tag{12}$$

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) = -\frac{1}{\sigma_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t), \tag{13}$$

in which $\mathbf{x}_t = \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}$ with α_t/σ_t the signal-to-noise ratio, and $\boldsymbol{\epsilon}_\theta$ the ground-truth noise prediction DPM. However, the implementation of classifier-free guidance in DPMs employs *two* evaluations of $\boldsymbol{\epsilon}_\theta$ at each single denoising step, doubling the time cost during inference. As a comparison, `GANdance` barely slow down the inference speed because of light-weight G_{map} module.

Table 1: Quantitative results of the conditional generation using Aurora [64] on ImageNet 64 x 64 dataset [7]. We calculate the FID score by drawing 50K samples with guidance strength $\lambda \in \{1.1, 1.2, 1.3\}$ with different sizes of the opposite condition set $\mathcal{C}$. The values in each cell represent the FID score relative to increasing λ.

Metric	$\lvert\mathcal{C}\rvert = 10$	$\lvert\mathcal{C}\rvert = 100$	$\lvert\mathcal{C}\rvert = 500$	$\lvert\mathcal{C}\rvert = 1000$
FID	7.16/6.43/6.20	7.07/6.27/6.17	6.98/6.16/6.07	6.92/6.15/6.06

4 Experiments

4.1 Experimental Setups

Datasets and baselines. We apply the proposed `GANdance` to previous seminal conditional GANs, including StyleGAN2 [26], BigGAN [4], and Aurora [64]. We train all three models on the ImageNet dataset [7] with different resolutions, *i.e.*, we introduce both 128x128 and 64x64 resolutions to StyleGAN, while using 128x128 and 64x64 on BigGAN and Aurora, respectively.
Evaluation metrics. We draw 50,000 samples for Fréchet Inception Distance (FID) [13] to evaluate the fidelity of the synthesized images. In addition, we use Improved Precision and Recall [28] to separately measure sample fidelity (Precision) and diversity (Recall).
Implementation details. We train `GANdance` using PyTorch [35] with NVIDIA Tesla A100 GPUs. We use the third-party implementation of StyleGAN2[1] [26] under Hammer [47] and officially implemented BigGAN[2] [4] and Aurora[3] [64].

4.2 Results of Training-Free `GANdance`

Recall that we introduce the training-free `GANdance` by averaging all potential conditions over a randomly sampled opposite condition set. Tab. 1 shows the effectiveness of our method, and we surprisingly observe that the performance degradation is inconspicuous with even a small opposite condition set.

4.3 Results of Learning-Based `GANdance`

Qualitative results. We show some visualization results in Figs. 2 and 3, by introducing learning-based guidance, our method has demonstrated good performance on both the 64-resolution and 128-resolution ImageNet datasets. We also show some visualization results in Fig. 5, in the first two rows, we see given "suit" and "trench coat" as class labels, original conditional generation can only output such bad cases. However, while our approach is applied and the guidance strength is gradually increased, the image quality improved significantly. Especially, in the case of "suit", guided sampling leads the model to the correct data distribution

[1] https://github.com/bytedance/Hammer
[2] https://github.com/ajbrock/BigGAN-PyTorch
[3] https://github.com/zhujiapeng/Aurora

Fig. 2: Diverse results generated by learning-based `GANdance` upon Aurora [64] on ImageNet 64x64 dataset [7]. We randomly sample eight global latent codes $\mathbf{z}$ for each label condition c, demonstrated in each row.

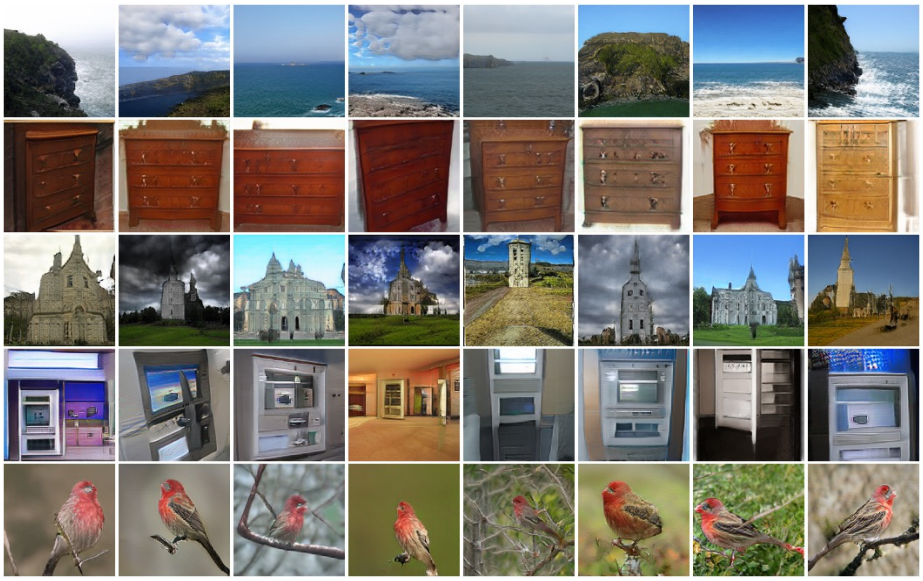

Fig. 3: Diverse results generated by learning-based `GANdance` upon BigGAN [4] on ImageNet 128x128 dataset [7]. We randomly sample eight global latent codes $\mathbf{z}$ for each label condition c, demonstrated in each row.

Table 2: Sample quality on ImageNet [7] with 64 x 64 and 128 x 128 resolutions. †Methods that utilize distillation techniques. ‡Methods that are trained by ourselves with official implementation. *Methods that are finetuned by ourselves with official implementation using `GANdance`. For clearer demonstration, one-step approaches are highlighted by gray color. Number of model parameters is also reported except for methods that utilize distillation techniques for more comprehensive comparison.

METHOD	# Parameter	NFE (↓)	FID (↓)	Precision (↑)	Recall (↑)
ImageNet 64x64					
PD† [42]	–	2	8.95	0.63	**0.65**
CD† [53]	–	2	**4.70**	**0.69**	0.64
PD† [42]	–	**1**	15.39	0.59	0.62
CD† [53]	–	**1**	6.20	0.68	0.63
ADM [8]	296M	250	**2.07**	**0.74**	0.63
EDM [22]	296M	79	2.44	0.71	**0.67**
DDIM [51]	296M	50	13.70	0.65	0.56
DPM-Solver [31]	296M	10	6.61	0.64	0.65
CT [53]	296M	2	11.10	0.69	0.56
CT [53]	296M	**1**	13.00	0.71	0.47
iCT [52]	269M	**1**	4.02	0.70	0.63
StyleGAN2‡ [26]	25M	**1**	21.32	0.42	0.36
StyleGAN2+`GANdance`	25M	**1**	17.80	0.58	**0.54**
Aurora‡ [64]	203M	**1**	8.87	0.41	0.48
Aurora+`GANdance`	203M	**1**	**4.37**	**0.73**	0.53
ImageNet 128x128					
ADM [8]	422M	250	**5.91**	**0.70**	0.65
DDIM [51]	422M	50	10.03	0.65	0.64
DPM-Solver [31]	422M	10	15.59	0.58	**0.67**
StyleGAN2‡ [26]	28M	**1**	25.39	0.52	0.51
StyleGAN2+`GANdance`	28M	**1**	19.63	0.59	**0.55**
BigGAN‡ [4]	70M	**1**	10.76	0.73	0.29
BigGAN+`GANdance` *	70M	**1**	**9.07**	**0.75**	0.32

from a wrong class. In the third and fourth rows, we can also observe that our method allows Aurora to generate correct data distribution according to the given category and improves the generation quality.

Quantitative results. Besides the exhibited qualitative results, we also compare quantitatively between baseline and `GANdance`-improving version on various state-of-the-art conditional GANs. In Tab. 2, we report the evaluation results with the number of model parameters on ImageNet for a more comprehensive comparison. We can tell that `GANdance` significantly facilitates the fidelity on the conditional generation task. The dramatic and steady improvement of the reported FID score across all models and datasets strongly confirms the correctness and effectiveness of our theory. It is also noteworthy that the utilization of `GANdance` on Aurora achieves superior generation performance compared to well-known Consistency Models (Consistency Distillation, CD) [53] even with NFE = 2, appearing comparable with state-of-the-art iCT [52]. In addition, to verify that our method is compatible with the truncation trick, we use these two methods to sample the model obtained by a learning-based `GANdance`. The

Table 3: Comparison Results of combining our method with truncation trick. We sample and calculate FID according to different parameter settings with a pre-trained Aurora on ImageNet 64 x 64. Each column increases the truncation strength ψ from top to bottom, and each row increases the guidance strength λ from left to right.

FID ↓	$\lambda = 1.0$	$\lambda = 1.1$	$\lambda = 1.2$	$\lambda = 1.3$	$\lambda = 1.4$
$\psi = 1.0$	6.62	5.26 (-1.07)	4.61 (-2.01)	4.48 (-2.14)	4.53 (-2.09)
$\psi = 0.9$	5.55 (-1.07)	4.61 (-2.01)	4.13 (-2.49)	**4.07 (-2.55)**	4.21 (-2.41)
$\psi = 0.8$	5.22 (-1.40)	4.47 (-2.15)	4.33 (-2.29)	4.32 (-2.30)	4.55 (-2.07)

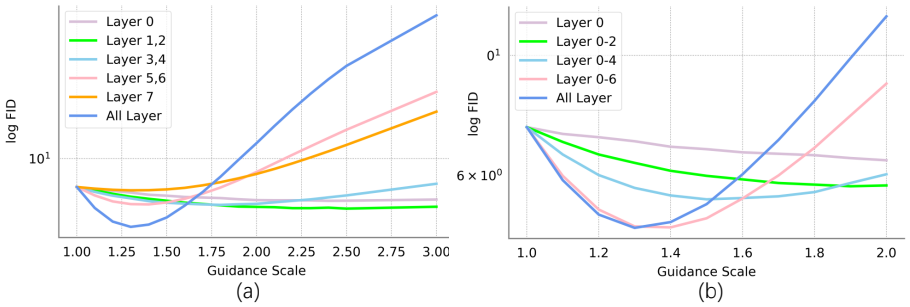

Fig. 4: Quantitative comparison measured by log FID (↓) of layer-wise guidance under different strength using Aurora [64] trained on ImageNet 64 x 64 [7]. (a) Applying guidance on each group of layers in the mapping network of Aurora. (b) Applying guidance on the first several groups of layers in the mapping network of Aurora.

results in Tab. 3 show that compared to the truncation trick, our method can significantly improve sample fidelity, and combining the two methods can obtain a more satisfactory result (*i.e.*, with FID 4.07).

Computational cost comparison. As one of the representative one-step generation paradigms, CD [53] distills the intricate knowledge to a new DPM model. Despite achieving respectable performance, the distillation process can be extremely computationally expensive. As reported in [53], CD involves 64 A100 GPUs for distillation with 600k iterations. As a comparison, training Aurora [64] from scratch with `GANdance` needs only 16 A100 GPUs and less than 300k iterations, even surpassing the performance of CD.

4.4 Results on Layer-Wise Guidance

Since GAN itself has a well-disentangled latent space, we apply our method to different parts of $\mathcal{W}$ space and analyze the results. We first apply guidance on each group of layers of Aurora, where the output features of layers in the same group have the same resolution. As shown in Fig. 4a, we see that the performance of the model applied guidance among all layers drops quickly when the guidance strength is larger than 1.3. The same phenomenon occurs in those models that applied guidance in 5, 6 (means 32x32 resolution) or 7 (means 64x64

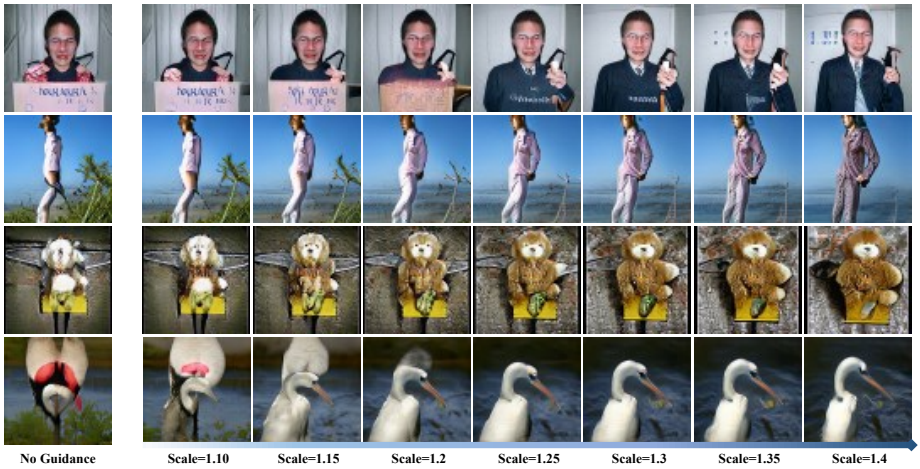

Fig. 5: Qualitative Results of the effect of different guidance scales. The leftmost column in the figure is the result of direct conditional generation by the model without any guidance techniques. The conditions given from top to bottom are: "suit", "trench coat", "teddy bear" and "American egret". The right part of the figure is the result of sampling with increasing guidance scale.

resolution). In contrast, increasing the guidance strength at the network level below 16x16 resolution has excellent performance. Recall that the channels of $\mathbf{w}$ controlling front layers (low resolution) of G are mainly responsible for high-level semantic information. The class label is also high-level semantic information for the picture itself. This also explains why guidance works well at small resolutions, while excessive guidance will cause network performance to collapse at large resolutions. The same phenomenon is also reflected in the experimental results of Fig. 4b. When we only operate at layers below 16x16 resolution, even if the guidance scale exceeds 1.5, the network performance still does not collapse.

4.5 Ablation Study

Guidance scale. Intuitively, the guidance strength λ is very important for `GANdance`. In more detail, small λ weakens the effectiveness of guidance, suggesting inconspicuous improvement. However, too large λ may drift the latent codes outside the reasonable region, harming the generation performance contrarily. We conduct a comprehensive ablation study to convey a direct and clear picture of the potential interval of guidance strength. As demonstrated in Fig. 6, the performance trend is consistent with the conclusion above.

Probability. Recall that in Sec. 3.3, we theoretically analyze the probability of resetting the condition as empty will influence the performance of the underlying GANs. As reported in Fig. 6, too large a probability will weaken the model performance. 0.1-0.2 is a suitable range.

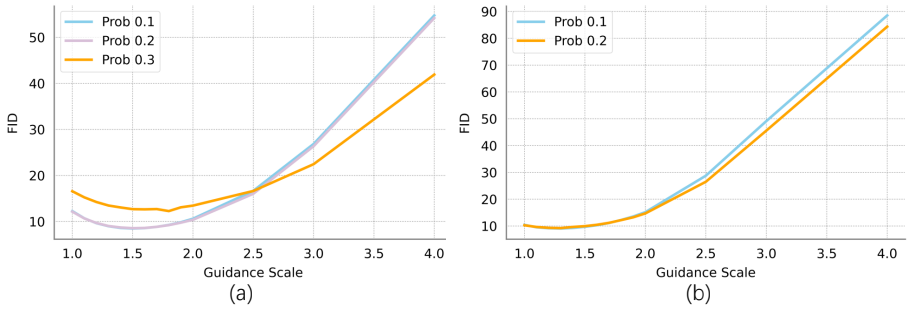

Fig. 6: Quantitative comparison of conditional generation performance of utilizing different probability to reset the condition to the newly added one using (a) Aurora [64] trained on ImageNet 64x64 dataset [7]; (b) BigGAN [4] trained on ImageNet 128x128 dataset [7]. We plot the FID score under different probability settings shown in different colors, in which the horizontal axis represents the guiding scale.

4.6 Discussion

It is widely recognized that GANs demonstrate poor capacity for handling multi-modal distribution and conditional generation, leaving GANs lacking further research such as text-to-image synthesis. Therefore, we believe our `GANdance` could inspire more findings and further encourage the development of a powerful GAN paradigm. Despite the great success of facilitating the fidelity of conditional generation, our proposed algorithm has several potential limitations. As a supplemental guidance, its efficacy depends highly on the strength λ. Furthermore, the probability of resetting the condition to the newly added empty one may influence the stability of native GANs. Although we conduct extensive and convincing ablation studies, the optimal strategy is still unexplored. Therefore, determining an adequate guidance strength and resetting probability according to different model settings and data domains will be an interesting.

5 Conclusion

In this paper, we introduce the guided sampling algorithm to GANs by delving into the background Bayesian foundation of classifier guidance. Drawing lessons from the well-organized latent space, we first point out the training-free version of `GANdance`, which estimates the joint distribution to provide effective classifier guidance in a plug-in fashion, avoiding the involvement of a classifier. This easy implementation brings dramatic performance improvement, barely increasing the inference time. By leveraging an *empty condition* attached with the entire dataset for GANs, we then propose a learning-based variant of `GANdance` to better approximate the distribution of the entire dataset, leading to further improvement. We conduct comprehensive experiments to demonstrate the efficacy of our method on a variety of baseline models.

Acknowledgements

This work was supported in part by the National Key R&D Program of China under Grant 2022YFA1005000, in part by the NSFC under Grant 61701304.

References

1. Abdal, R., Qin, Y., Wonka, P.: Image2stylegan: How to embed images into the stylegan latent space? In: ICCV (2019)
2. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: ICML (2017)
3. Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y., et al.: Improving image generation with better captions. Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf **2**(3), 8 (2023)
4. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. In: ICLR (2019)
5. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., De Mello, S., Gallo, O., Guibas, L.J., Tremblay, J., Khamis, S., et al.: Efficient geometry-aware 3d generative adversarial networks. In: CVPR (2022)
6. Chan, E.R., Monteiro, M., Kellnhofer, P., Wu, J., Wetzstein, G.: pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In: CVPR (2021)
7. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
8. Dhariwal, P., Nichol, A.Q.: Diffusion models beat GANs on image synthesis. In: NeurIPS (2021)
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. NeurIPS (2014)
10. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS (2014)
11. Gu, J., Liu, L., Wang, P., Theobalt, C.: Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. arXiv preprint arXiv:2110.08985 (2021)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
13. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: NeurIPS (2017)
14. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS. pp. 6840–6851 (2020)
15. Ho, J., Salimans, T.: Classifier-free diffusion guidance. In: Adv. Neural Inform. Process. Syst. Worksh. (2021)
16. Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., Fleet, D.J.: Video diffusion models. arXiv preprint arXiv:2204.03458 (2022)
17. Hu, L., Gao, X., Zhang, P., Sun, K., Zhang, B., Bo, L.: Animate anyone: Consistent and controllable image-to-video synthesis for character animation. arXiv preprint arXiv:2311.17117 (2023)
18. Huang, L., Chen, D., Liu, Y., Yujun, S., Zhao, D., Jingren, Z.: Composer: Creative and controllable image synthesis with composable conditions. arXiv preprint arxiv:2302.09778 (2023)

19. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR (2017)
20. Kang, M., Zhu, J.Y., Zhang, R., Park, J., Shechtman, E., Paris, S., Park, T.: Scaling up gans for text-to-image synthesis. arXiv preprint arXiv:2303.05511 (2023)
21. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. In: ICLR (2018)
22. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. In: NeurIPS (2022)
23. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. In: NeurIPS (2021)
24. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: CVPR (2019)
25. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: CVPR (2020)
26. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: CVPR. pp. 8107–8116 (2020)
27. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014)
28. Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., Aila, T.: Improved precision and recall metric for assessing generative models. arXiv preprint arXiv:1904.06991 (2019)
29. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR (2017)
30. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: ICCV (2015)
31. Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In: NeurIPS (2022)
32. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
33. Nguyen-Phuoc, T., Li, C., Theis, L., Richardt, C., Yang, Y.L.: Hologan: Unsupervised learning of 3d representations from natural images. In: ICCV (2019)
34. Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al.: Conditional image generation with pixelcnn decoders. In: NeurIPS (2016)
35. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: NeurIPS (2019)
36. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 (2022)
37. Rezende, D., Mohamed, S.: Variational inference with normalizing flows. In: ICML (2015)
38. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. CVPR (2021)
39. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10684–10695 (2022)
40. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22500–22510 (2023)
41. Saito, M., Matsumoto, E., Saito, S.: Temporal generative adversarial nets with singular value clipping. In: ICCV (2017)

42. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. In: ICLR (2022)
43. Sauer, A., Karras, T., Laine, S., Geiger, A., Aila, T.: Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis. arXiv preprint arXiv:2301.09515 (2023)
44. Sauer, A., Schwarz, K., Geiger, A.: Stylegan-xl: Scaling stylegan to large diverse datasets. In: SIGGRAPH (2022)
45. Schwarz, K., Liao, Y., Niemeyer, M., Geiger, A.: Graf: Generative radiance fields for 3d-aware image synthesis. In: NeurIPS (2020)
46. Shen, Y., Gu, J., Tang, X., Zhou, B.: Interpreting the latent space of gans for semantic face editing. In: CVPR (2020)
47. Shen, Y., Zhang, Z., Yang, D., Xu, Y., Yang, C., Zhu, J.: Hammer: An efficient toolkit for training deep models. `https://github.com/bytedance/Hammer` (2022)
48. Shen, Y., Zhou, B.: Closed-form factorization of latent semantics in gans. In: CVPR (2020)
49. Shi, Z., Peng, S., Xu, Y., Geiger, A., Liao, Y., Shen, Y.: Deep generative models on 3d representations: A survey. arXiv preprint arXiv:2210.15663 (2022)
50. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: ICML. pp. 2256–2265 (2015)
51. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: ICLR (2021)
52. Song, Y., Dhariwal, P.: Improved techniques for training consistency models. arXiv preprint arXiv:2310.14189 (2023)
53. Song, Y., Dhariwal, P., Chen, M., Sutskever, I.: Consistency models. arXiv preprint arXiv:2303.01469 (2023)
54. Tao, M., Bao, B.K., Tang, H., Xu, C.: Galip: Generative adversarial clips for text-to-image synthesis. In: CVPR (2023)
55. Tulyakov, S., Liu, M.Y., Yang, X., Kautz, J.: Mocogan: Decomposing motion and content for video generation. In: CVPR (2018)
56. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. In: NeurIPS (2017)
57. Wu, Z., Lischinski, D., Shechtman, E.: Stylespace analysis: Disentangled controls for stylegan image generation. In: CVPR (2021)
58. Xu, Y., Peng, S., Yang, C., Shen, Y., Zhou, B.: 3d-aware image synthesis via learning structural and textural representations. In: CVPR (2022)
59. Yang, C., Shen, Y., Zhou, B.: Semantic hierarchy emerges in deep generative representations for scene synthesis. IJCV (2020)
60. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: CVPR (2018)
61. Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-attention generative adversarial networks. In: ICML (2019)
62. Zhu, J., Yang, C., Shen, Y., Shi, Z., Dai, B., Zhao, D., Chen, Q.: Linkgan: Linking GAN latents to pixels for controllable image synthesis. In: ICCV (2023)
63. Zhu, J., Yang, C., Shen, Y., Shi, Z., Dai, B., Zhao, D., Chen, Q.: Linkgan: Linking gan latents to pixels for controllable image synthesis. In: ICCV (2023)
64. Zhu, J., Yang, C., Zheng, K., Xu, Y., Shi, Z., Shen, Y.: Exploring sparse MoE in GANs for text-conditioned image synthesis. arXiv preprint arXiv:2309.03904 (2023)