

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2023.0322000

Latent Denoising Diffusion GAN: Faster sampling, Higher image quality

LUAN THANH TRINH¹, TOMOKI HAMAGAMI²

¹Department of Mathematics, Physics, Electrical Engineering and Computer Science, Yokohama National University (e-mail: tt.luan.ynu@gmail.com)

²Department of Mathematics, Physics, Electrical Engineering and Computer Science, Yokohama National University (e-mail: hamagami@ynu.ac.jp)

ABSTRACT Diffusion models are emerging as a powerful solution for generating high-fidelity and diverse images, often surpassing GANs under many circumstances. However, their slow inference speeds hinder their potential for real-time applications. To address this, DiffusionGAN leveraged a conditional GAN to drastically reduce denoising steps and speed up inference. Its advancement, Wavelet Diffusion, further accelerated the process by converting data into wavelet space, thus enhancing efficiency. Nonetheless, these models still fall short of GANs in terms of speed and image quality. To bridge these gaps, this paper introduces the Latent Denoising Diffusion GAN, which employs pre-trained autoencoders to compress images into a compact latent space, significantly improving inference speed and image quality. Furthermore, we propose a Weighted Learning strategy to enhance diversity and image quality. Experimental results on the CIFAR-10, CelebA-HQ, and LSUN-Church datasets prove that our model achieves a state-of-the-art running speed among diffusion models. Compared to its predecessors, DiffusionGAN and Wavelet Diffusion, our model shows remarkable improvements on all evaluation metrics. Code and pre-trained checkpoints: <https://github.com/thanhluantrinh/LDDGAN.git>.

INDEX TERMS Diffusion models, Image generation, Generative adversarial networks

I. INTRODUCTION

Despite being a recent introduction, diffusion models have quickly established themselves as a pivotal paradigm for image-generation tasks. At their core, diffusion models hinge on two crucial processes: the forward process (or diffusion process) and the reverse process (denoising process). In the forward process, Gaussian noise is incrementally infused into the input image until it transforms into an isotropic Gaussian. Conversely, in the reverse process, a model is meticulously trained to invert the forward process and faithfully reproduce the original input image. After training, the power of diffusion models shines as we can generate high quality images by navigating randomly sampled noise through the adeptly learned denoising process.

In comparison to other prominent deep generative models, diffusion models distinguish themselves through the excellence of the generated images in terms of quality and diversity, coupled with their inherent training stability. Particularly noteworthy is the observation that diffusion models have surpassed Generative Adversarial Networks (GANs), which have dominated the image generation task in recent years, excelling in both image quality ([1], [2]) and diversity ([3], [4], [49]). One notable aspect heightening expectations for diffusion models is their growing capacity to effectively in-

corporate various conditional inputs, such as semantic maps, text, representations, and images. This versatility expands the potential applications of diffusion models into areas like text-to-image generation ([2], [5], [30]), video generation ([8], [10]), image-to-image translation ([6], [7], [9]), text-to-3D generation [11], and beyond.

Despite their considerable potential, the hindrance of slow inference speed poses a significant obstacle for diffusion models in becoming fully-fledged image generation models that can meet the diverse expectations across various domains, particularly in real-time applications. The fundamental cause of slow sampling in these models lies in the Gaussian assumption made during the denoising step, a assumption that is valid only for small step sizes. Consequently, diffusion models often necessitate a substantial number of denoising steps, typically ranging in the hundreds or thousands. By modeling complex and multimodal distributions through conditional GANs, DiffusionGAN [28] enables larger denoising steps, reducing the number of denoising steps to just a few, thereby significantly accelerating the inference time of diffusion models. Wavelet Diffusion [29], an enhancement of DiffusionGAN, achieves a further increase in inference speed by transferring data from pixel space to wavelet space,

reducing the input data size by a factor of 4 and becoming the fastest existing diffusion model. However, Wavelet Diffusion still lags considerably behind StyleGAN [60]. Additionally, the acceleration of inference speed shows signs of compromising the output image quality, as the output quality of both DiffusionGAN and Wavelet Diffusion is lower than that of StyleGAN and recent diffusion models.

This paper aims to bridge both the gap in image quality and the gap in speed by introducing Latent Denoising Diffusion GAN (LDDGAN). Firstly, instead of residing in a high-dimensional pixel space, input images are compressed as much as possible into a low-dimensional latent space through pre-trained autoencoders. This compression significantly reduces computational costs during both training and inference, facilitating faster sampling. Given that the latent space is more suitable for diffusion models than the high-dimensional pixel space [30], our approach aims to enhance both image quality and sample diversity by utilizing this space for both the diffusion and denoising processes. Following the principles of DiffusionGAN, a conditional GAN is employed to model complex and multimodal distributions and enable a large denoising step. Additionally, to enhance diversity through adversarial loss while leveraging the effect of reconstruction loss for image quality improvement, we propose a novel learning strategy called Weighted Learning. Since our model primarily uses multimodal distributions, instead of restricting latents learned by the autoencoder to Gaussian distributions as in other latent-based models ([13], [30], [45]), we allow the autoencoder to freely search for the appropriate latent space. This helps to significantly accelerate the convergence of the autoencoder and improve the overall quality and diversity of the main model.

Based on experimental results on standard benchmarks including CIFAR-10, CELEBA-HQ, and LSUN Church, our model achieves a state-of-the-art running speed for a diffusion model while maintaining high image quality. When compared to GANs, we achieve comparable image generation quality and running speed, along with increased diversity. Additionally, in comparison to two predecessors, DiffusionGAN and Wavelet Diffusion, our model significantly outperforms them across the majority of comparison metrics.

In summary, our contributions are as follows:

- We propose a novel Latent Denoising Diffusion GAN framework that leverages dimensionality reduction and the high compatibility of low-dimensional latent spaces with the diffusion model's denoising process. This approach not only improves inference speed and the quality of generated images but also enhances diversity.
- We find that if the denoising process of a diffusion model is not dependent on Gaussian distributions, it becomes necessary to remove the autoencoder's learned space dependence on the Gaussian distribution to enhance the diversity and quality of the generated images. This insight could serve as a recommendation for future latent-based diffusion models.

- We propose an innovative Weighted Learning strategy that boosts diversity via adversarial loss, while also improving image quality through the effect of reconstruction loss.
- Our Latent Denoising Diffusion GAN features low training costs and state-of-the-art inference speeds, paving the way for real-time and high-fidelity diffusion models.

II. RELATED WORK

A. IMAGE GENERATION MODELS

GANs (Generative Adversarial Networks, [12]) are among the representative generative models extensively utilized in various real-time applications due to their ability to generate high-quality images rapidly [17]. However, challenges in optimization ([16], [18], [19]) and the difficulty in capturing the full data distribution [20] are two main weaknesses of GANs. In contrast to GANs, Variational autoencoders (VAE, [13]) and flow-based models ([14], [15]) excel in fast inference speed and high sample diversity but face challenges in generating images of high quality. Recently, diffusion models have been introduced and have quickly made an impressive impact by achieving state-of-the-art results in both sample diversity [4] and sample quality [21]. One of the two primary weaknesses of diffusion models, high training costs, has been effectively addressed by Latent Diffusion Models (LDMs, [30]). This was achieved by shifting the diffusion process from the pixel space to the latent space using pre-trained autoencoders. Despite the similarity between this approach and ours in using latent space, their reliance on Gaussian distributions in both the denoising process and the latent space learned by the autoencoder necessitates thousands of network evaluations during the sampling process. This dependency results in slow inference speeds, representing the primary remaining weakness of diffusion models. Consequently, it becomes a significant bottleneck, restricting the practical use of diffusion models in real-time applications.

B. FASTER DIFFUSION MODELS

To enhance the inference speed of diffusion models, several methods have been proposed, including learning an adaptive noise schedule in [22], using non-Markovian diffusion processes in [47], [55], and employing improved SDE solvers for continuous-time models in [23]. However, these methods either experience notable deterioration in sample quality or continue to necessitate a considerable number of sampling steps. Studies on knowledge distillation ([57], [69], [70]) are also worth noting as, in some cases, they can perform the diffusion model with just a few steps or even just one step and produce high-quality images. The weakness of knowledge distillation method is that it requires a well pre-trained diffusion model to be used as a teacher model for the main model (student model). The results of the student model often struggle to surpass the teacher model due to this student-teacher constraint.

The most successful method to date for accelerating inference speed without compromising image quality, and without

relying on another pre-trained diffusion model, is DiffusionGAN [28]. By modeling complex and multimodal distributions through conditional GANs, DiffusionGAN enables larger denoising steps, reducing the number of denoising steps to just a few and significantly accelerating the inference. Wavelet Diffusion [29], an enhancement of DiffusionGAN, achieves a further increase in inference speed by transferring data from pixel space to wavelet space, reducing the input data size by a factor of 4 and becoming the fastest existing diffusion model. However, the inference speed of Wavelet Diffusion is still much slower than that of GAN, and the trade-off between sampling speed and quality still requires further improvement. We aim to address both problems by utilizing latent space.

C. LATENT-BASED APPROACHES

Regarding diffusion model families, similar to our approach, LDMs in [30] and Score-based generative models (SGMs) in [45] utilize an autoencoder to transform data from pixel space to latent space and conduct the diffusion process in this space. However, as mentioned above, their dependence on Gaussian distributions in both the denoising process and the latent space learned by the autoencoder results in the necessity for thousands of network evaluations during the sampling process. This dependency leads to slow inference speeds. Our method enables larger denoising steps, thereby reducing the number of denoising steps to just a few and significantly accelerating the inference speed. Additionally, due to the absence of a reliance on Gaussian distribution, unlike LDMs and SGMs, our autoencoders are free to explore potential latent spaces rather than attempting to model spaces with Gaussian distribution. Details about our autoencoders are described in Sec. IV-B.

III. BACKGROUND

DIFFUSION MODELS

The main idea of Diffusion models ([30], [45], [47], [49]) lies in gradually adding noise to the data and then training a model to reverse that process gradually. In this way, the model becomes capable of generating data similar to the input data from pure Gaussian noise. The forward process, in which we gradually add noise to the input data $\mathbf{x}_0$, can be defined by:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t \geq 1} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad (1)$$

with $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t}, \beta_t \mathbf{I})$

where T and β_t denote number of steps and pre-defined variance schedule at timesteps t , respectively. The reverse process, in which we remove noise from data, can be defined:

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t \geq 1} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \quad (2)$$

with $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \sigma_t^2 \mathbf{I})$

where $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$ and σ_t^2 are the mean and variance for the denoising model parameterized by θ . As introduced in [27], the optimal mean can be defined as follows.

$$\boldsymbol{\mu}(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \alpha_t}} \mathbb{E}[\epsilon|\mathbf{x}_t] \right) \quad (3)$$

Here, $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$, $\alpha_t = 1 - \beta_t$ and ϵ is the noise added to $\mathbf{x}_t$. Therefore, learning object can be considered as a noise prediction task:

$$\min_{\theta} = \mathbb{E}_{t, \mathbf{x}_0, c, \epsilon} \|\epsilon - \epsilon_\theta(\mathbf{x}_t, t, c)\|_2^2 \quad (4)$$

with c is the condition fed into the model (e.g., class-conditional [45] or text- to-image [30]). Eq. 2 suggests that diffusion models often make the assumption that the denoising distribution can be approximated by Gaussian distributions. Nevertheless, it is widely acknowledged that the Gaussian assumption only holds in the infinitesimal limit of small denoising steps (as highlighted by Sohl-Dickstein *et al.* in [25] and Feller in [26]). This necessitates a large number of steps in the reverse process and causes a slow sampling issue for diffusion models.

DENOISING DIFFUSION GAN AND WAVELET DIFFUSION

To reduce the inference time of diffusion models, it's crucial to significantly decrease the number of denoising diffusion steps T required for the reverse process. Additionally, a new denoising method is required because using large denoising steps causes the denoising distributions $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ to become more complex and multimodal, unlike the existing diffusion models ([30], [45], [47]) where the denoising distributions follow a Gaussian distribution. To overcome this challenge, DiffusionGAN models the denoising distribution with an complex and multimodal distribution by using conditional GANs. Whereas existing diffusion models predict noise ϵ added to $\mathbf{x}_{t-1}$ at timestep t using $\mathbf{x}_t$, the generator $G_\theta(\mathbf{x}_t, z, t)$ in DiffusionGAN predicts the input data $\mathbf{x}_0$ with random latent variable $z \sim \mathcal{N}(0, \mathbf{I})$. This crucial distinction allows DiffusionGAN's denoising distribution $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ to become multimodal and complex, unlike the unimodal denoising model of existing approaches. The perturbed sample $\mathbf{x}'_{t-1}$ is then acquired using pre-defined $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$. The discriminator D_ϕ performs judgment on fake pairs $D_\phi(\mathbf{x}'_{t-1}, \mathbf{x}_t, t)$ and real pairs $D_\phi(\mathbf{x}_{t-1}, \mathbf{x}_t, t)$.

Building upon Diffusion GAN, Wavelet Diffusion further speeds up inference by performing denoising in wavelet space rather than pixel space. It first decomposes the input image $x \in \mathbb{R}^{3 \times H \times W}$ into its low-frequency and high-frequency components using a wavelet transformation and concatenates them to form a matrix $y \in \mathbb{R}^{12 \times \frac{H}{2} \times \frac{W}{2}}$. This helps to reduce the input data size by 4 times and decrease the inference time of the model as well as the computational cost. To better exploit the features and improve the output image quality, Wavelet Diffusion proposes a wavelet-embedded generator optimized for wavelet space. Wavelet Diffusion also introduces a reconstruction loss between the generated data and its ground truth to boost the model training convergence.

In the following sections, DDGAN and WDDGAN will stand for DiffusionGAN and Wavelet Diffusion, respectively, for brevity.

IV. METHOD

In this section, we begin by providing a comprehensive overview of the proposed framework, Latent Denoising Diffusion GAN (Sec. IV-A). Following this, we introduce the autoencoder utilized to train the main model (Sec. IV-B). Finally, we discuss the effect of reconstruction loss on the final result and introduce a novel training strategy called Weighted Learning (Sec. IV-C).

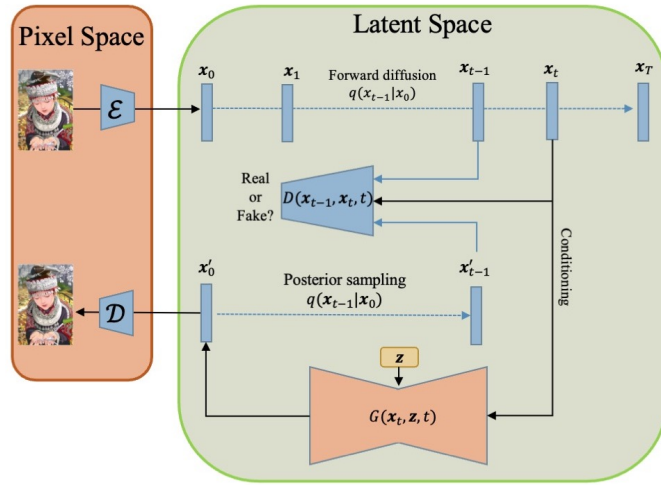


FIGURE 1: The training process of Latent Denoising Diffusion GAN (LDDGAN)

A. LATENT DENOISING DIFFUSION GAN

Figure 1 provides an overview of our proposed LDDGAN. A pre-trained Encoder is used to compress input data X_0 and transform it from pixel space to low-dimensional latent representation $x_0 = \mathcal{E}(X_0)$. It's important to note that the encoder downsamples the image by a factor f . We strive to compress the data as much as possible (larger f), while still guaranteeing image quality upon decoding. Details about the autoencoder are described in Section IV-B.

The forward diffusion process and reverse process are performed on low-dimensional latent space instead of wavelet space as in WDDGAN, or pixel space as in DDGAN and inherent diffusion models. This method has two main advantages. Firstly, LDDGAN is capable of reducing the input image size by 4, 8 times, or even more, unlike WDDGAN which can only reduce it by 4 times. This helps to enhance the model's inference speed and significantly reduce computational costs. As investigated by Rombach *et al.* in [30], compared to the high-dimensional pixel space, the low-dimensional latent space is better suited for the diffusion model, a likelihood-based generative method. This results in improved quality and diversity of the generated samples.

In the latent space, to reduce inference time, the forward diffusion process is performed with a small number of sampling steps T ($T \leq 8$), and each diffusion step has a significantly larger β_t compared to the traditional diffusion models. At timestep t , a corrupted sample x_t is generated from the input sample x_0 using the posterior distribution $q(x_{t-1}|x_0)$. The generator $G(x_t, z, t)$ uses x_t , the current timestep t , and a random latent variable $z \sim \mathcal{N}(0, \mathbf{I})$ to predict an approximation of the original sample x'_0 . The predicted sample x'_t is then calculated using the same posterior distribution $q(x_{t-1}|x_0)$. Finally, the discriminator evaluates both the fake pairs $D(x'_{t-1}, x_t, t)$ and real pairs $D(x_{t-1}, x_t, t)$ to provide feedback to the generator. The decoder $\mathcal{D}$ is not necessary for training but is used to convert x'_0 to the output image. We describe the sampling process in Algorithm 1.

Algorithm 1 Sampling process of LDDGAN

```

 $x_T \sim \mathcal{N}(0, \mathbf{I})$ 
for  $t = T, \dots, 1, 0$  do
     $z \sim \mathcal{N}(0, \mathbf{I})$ 
     $x'_0 = G(x_t, z, t)$ 
     $x'_{t-1} \sim q(x_{t-1}|x_0)$ 
end for
 $X_0 = \mathcal{D}(x'_0)$ 
return  $X_0$ 

```

B. AUTOENCODER

In the context of latent space representation learning, recent studies, particularly diffusion models ([30], [33], [34]) and variational autoencoders (VAEs, [31], [32]), frequently employ KL-penalty (Kullback-Leibler divergence, [53], [54]) between Gaussian distribution and learned latent within the loss function. This approach encourages the learned latent space to closely approximate a Gaussian distribution, proving effective when model assumptions rely heavily on Gaussian distributions. However, with models that allow complex and multimodal distributions, as in this study, we do not use this KL-penalty. Instead, we let our autoencoder freely exploit latent spaces and prioritize the ability to compress and recover images.

We construct our autoencoder architecture based on VQGAN [35], distinguished by the integration of the quantization layer within the decoder. To ensure adherence to the image manifold and promote local realism in reconstructions, we employ a training regimen encompassing both perceptual loss [36] and a patch-based adversarial loss ([37], [38]) similar to VQGAN. This approach effectively circumvents the potential for blurriness that can arise when training relies exclusively on pixel-space losses such as L2 or L1 objectives. Furthermore, motivated by investigations in [30] that have demonstrated the superiority of two-dimensional latent variables over traditional one-dimensional counterparts in the domains of image compression and reconstruction, we elected to utilize a two-dimensional latent variable as both the output of our encoder and the input to the subsequent

diffusion process.

C. LOSS FUNCTIONS

1) Adversarial loss

The adversarial loss for the discriminator and the generator are defined as follows.

$$\mathcal{L}_{adv}^D = -\log(D(x_{t-1}, x_t, t)) + \log(D(x'_{t-1}, x_t, t)) \quad (5)$$

$$\mathcal{L}_{adv}^G = -\log(D(x'_{t-1}, x_t, t)) \quad (6)$$

2) Reconstruction loss and Weighted Learning

In WDDGAN [29], in addition to the adversarial loss, a reconstruction loss between the generated sample and its ground truth is also used when training the generator. The generator's overall objective function is mathematically formulated as a linear combination of adversarial loss and reconstruction loss, weighted by a fixed hyperparameter, expressed as follows.

$$\mathcal{L}_{rec}^G = ||x_0 - x'_0|| \quad (7)$$

$$\mathcal{L}_G = \mathcal{L}_{adv}^D + \lambda \mathcal{L}_{rec}^G \quad (8)$$

As investigated in [29], using reconstruction loss helps WDDGAN improve image fidelity. This can be explained by the fact that when only adversarial loss is used, the generator learns to produce x'_0 that is similar to the reference input x_0 indirectly, through feedback from the discriminator. However, reconstruction loss provides the generator with direct feedback through the similarity between x_0 and x'_0 , calculated using L1 loss. Therefore, using reconstruction loss in conjunction with adversarial loss helps to make the training of the generator easier and achieve better convergence. There is also evidence in [29] that WDDGAN converges faster than DDGAN. However, we postulate that simply employing a linear combination of the two loss functions, as presented in Eq. 8, to construct the overall loss function proves ineffective and potentially reduces the diversity of generated images. This hypothesis stems from two primary considerations. Firstly, at a certain training stage, the generator acquires the ability to produce data closely resembling the input data solely through the guidance of the discriminator's feedback, effectively negating the necessity of reconstruction loss. Secondly, reconstruction loss may cause the generator to tend to generate data that is identical to the training data for any input noise, thereby restricting its capacity for creative exploration and diverse generation.

In this study, we also apply reconstruction loss. However, instead of a simple linear combination, we propose a method called Weighted Learning to combine the overall loss function of the generator as follows.

$$\mathcal{L}_G = \mathcal{L}_{adv}^D + \lambda \mathcal{L}_{rec}^G \quad (9)$$

$$\text{with } \varphi = -\delta + \delta \frac{\text{Epoch}}{\text{NumEpoch}} \quad \text{and} \quad \lambda = 1 - \frac{1}{1 + \exp^{-\varphi}}$$

Here, δ is a weighting hyper-parameter and Figure. 2 is an example of our method. In the early stages of training, the importance of reconstruction loss remains largely unchanged with $\lambda \simeq 1$. This helps to leverage reconstruction loss to mitigate the weaknesses of adversarial loss caused by indirect learning through the discriminator and increase the convergence speed of the model. In the next stage, the importance of reconstruction loss is reduced, giving priority to reconstruction loss. This is expected to help increase the diversity of generated images. Finally, when training is nearing completion, the importance of reconstruction loss is gradually reduced until it approaches 0 to avoid excessive changes in the overall training objective and increase training stability.

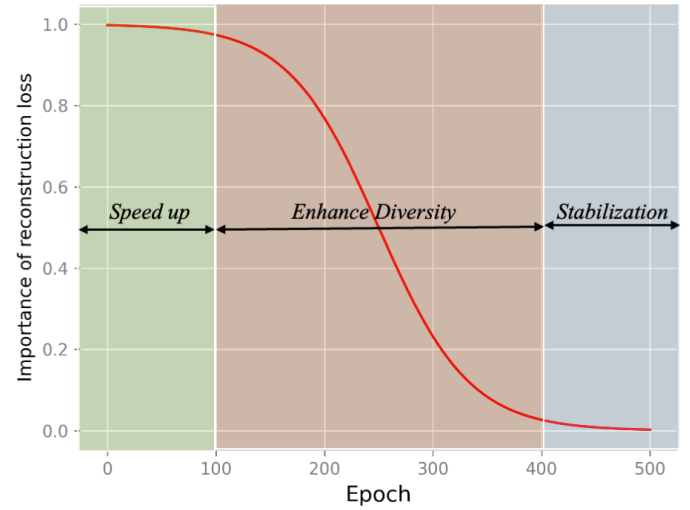


FIGURE 2: An example of Weighted Learning

Before starting with Weighted Learning, we also tried to remove reconstruction function after training the model with both loss functions to a certain extent. However, suddenly changing the training objective caused the model to become unstable and difficult to converge again. The experiments to verify the effectiveness of Weighted Learning are presented in Section V.

V. EXPERIMENTS

We first present a detailed description of the experimental settings, dataset, and evaluation metrics used in Section V-A. Then, we present experimental results that verify the effectiveness of the proposed LDDGAN and comparison results with previous studies in Section V-B. Finally, we evaluate the effectiveness of Weighted Learning and the effect of the learned latent space by the Autoencoder in Section V-C.

A. EXPERIMENTAL SETUP

1) Datasets

To save computational costs, we use CIFAR-10 32×32 as the main dataset for qualitative and quantitative comparisons with previous studies, as well as evaluating the effectiveness of each component added to LDDGAN. To further visualize

the effectiveness of the model, we use LSUN Church 256×256 [40] and CELEBA-HQ 256×256 [39] datasets to evaluate through the high-resolution image generation task.

2) Evaluation metrics

When evaluating our model, we consider three key factors: inference time, sample fidelity, and sample diversity. To assess inference time, we measure the number of function evaluations (NFE) and the average time taken to generate a batch size of 100 across 300 trials. We use the widely used Frechet inception distance (FID, by Husel *et al.* in [41]) metric for sample fidelity. To measure sample diversity, we rely on the improved recall score developed by Kynkaanniemi *et al.* in [42], which is the improved version of the original one proposed by Sajjadi *et al.* in [43]. For comparison, we use FID and Recall data from previously published papers. To ensure that the FID and Recall results remain unchanged compared to the original papers, we adopt the code and experimental environment from the published code of the previous studies mentioned above.

3) Autoencoder

As described in Sec. IV-B, we build our autoencoder based on VQGAN [35], distinguished by the integration of the quantization layer within the decoder. The autoencoder is trained using a perceptual loss [36] and patch-based adversarial loss ([37], [38]) to maintain adherence to the image manifold and promote local realism in the reconstructions. We try to compress data as much as possible (larger scale factor f), while still ensuring image quality when decoding. Table. 1 lists the autoencoders that were successfully trained for each dataset. All models were trained until convergence, which is when there was no further substantial improvement in FID.

Dataset	Scale factor f	Output size	FID
CIFAR-10 32×32	2	$16 \times 16 \times 4$	1.32
CELEBA-HQ 256×256	4	$64 \times 64 \times 4$	0.58
LSUN Church 256×256	8	$32 \times 32 \times 3$	1.14

TABLE 1: Successfully trained autoencoders for each dataset. FID is calculated between the reconstructed images after compression and the original images of the validation set. The output size of these autoencoders will be the input size of the discriminator and the generator.

4) Implementation details

Our implementation is based on DDGAN [28], and we followed the same training configurations as in DDGAN for our experiments. In constructing our GAN generator, we align with DDGAN's architectural choice by employing the U-Net-structured NCSN++ framework as presented in Song *et al.* [44]. To effectively model the denoising distribution within a complex and multimodal context, we leverage latent variable $\mathbf{z}$ to exert control over normalization layers. To achieve this, we strategically substitute all group normalization layers (Wu and He [46]) within the generator with adaptive group nor-

malization layers. This technique involves utilizing a straightforward multi-layer fully-connected network to accurately predict the shift and scale parameters within group normalization directly from $\mathbf{z}$. The network's input is constituted by the conditioning element $\mathbf{x}_t$, and time embedding is meticulously employed to guarantee conditioning on t .

Thanks to the efficient compression of input data through autoencoders, our models require fewer GPU resources compared to WDDGAN and DDGAN. With a batch size equal to or greater than, all of our models can be trained on just 1-2 NVIDIA RTX A5000 GPUs (24GB), whereas WDDGAN and DDGAN require 1-8 NVIDIA V100 GPUs (32 GB) or NVIDIA A100 GPUs (40 GB). Our model only requires a configuration similar to DDGAN's CIFAR10 model for high-resolution datasets LSUN and CELEBA. A detailed comparison of computational cost between DDGAN, WDDGAN, and our models is presented in Table 2. Our model has a comparable number of parameters but uses significantly less computing FLOPs and memory.

B. EXPERIMENTAL RESULTS

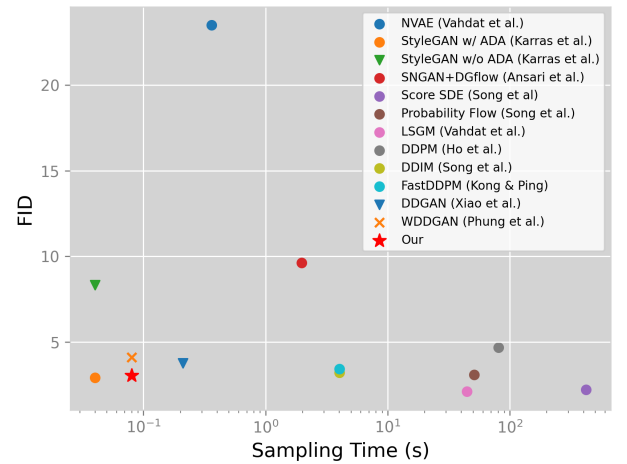


FIGURE 3: Sample quality vs sampling time trade-off.

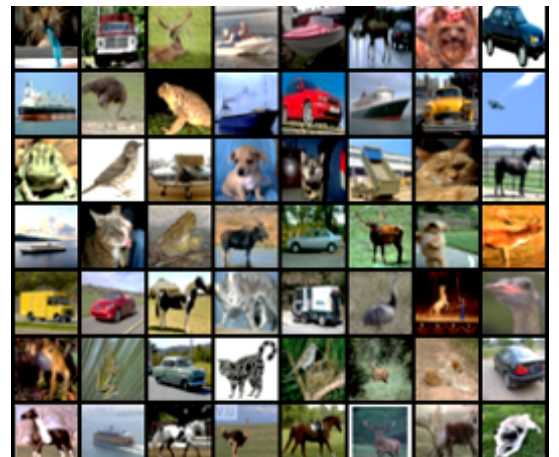


FIGURE 4: CIFAR-10 qualitative samples.

	DDGAN			WDDGAN			Ours (*)		
	Params	FLOPs	MEM	Params	FLOPs	MEM	Params	FLOPs	MEM
CIFAR-10 (32)	48.43M	7.05G	0.31G	33.37M	1.67G	0.16G	41.97M	1.72G	0.18G
CELEBA-HQ (256)	39.73M	70.82G	3.21G	31.48M	28.54G	1.07G	45.39M	7.15G	0.27G
LSUN(256)	39.73M	70.82G	3.21G	31.48M	28.54G	1.07G	40.92M	9.68G	0.25G

TABLE 2: Model specifications of DDGAN [28], WDDGAN [29] and our approach including a number of parameters (M), FLOPs (GB), and memory usage (GB) on a single GPU for one sample. (*) including the decoder.

Model	FID↓	Recall↑	NFE↓	Time(s)↓
Ours	2.98	0.58	4	0.08 (*)
WDGAN [29]	4.01	0.55	4	0.08
DDGAN [28]	3.75	0.57	4	0.21
DDP [47]	3.21	0.57	1000	80.5
NCSN [44]	25.3	-	1000	107.9
Adversarial DSM [48]	6.10	-	1000	-
Likelihood SDE [49]	2.87	-	1000	-
Probability Flow (VP) [50]	3.08	0.57	140	50.9
LSGM [51]	2.10	0.61	147	44.5
Score SDE (VE) [45]	2.20	0.59	2000	423.2
Score SDE (VP) [45]	2.41	0.59	2000	421.5
DDIM [52]	4.67	0.53	50	4.01
FastDDPM [55]	3.41	0.56	50	4.01
Recovery EBM [56]	9.58	-	180	-
DDPM Distillation [57]	9.36	0.51	1	-
SNGAN+DGflow [58]	9.62	0.48	25	1.98
AutoGAN [59]	12.4	0.46	1	-
StyleGAN2 w/o ADA [60]	8.32	0.41	1	0.04
StyleGAN2 w/ ADA [60]	2.92	0.49	1	0.04
StyleGAN2 w/ Diffaug [61]	5.79	0.42	1	0.04
Glow [62]	48.9	-	1	-
PixelCNN [63]	65.9	-	1024	-
NVAE [64]	23.5	-	0.51	0.36
VAEBM [65]	12.2	0.53	16	8.79

TABLE 3: Results on CIFAR-10. (*) including reference time of the decoder. Among diffusion models, our method attains a state-of-the-art speed while preserving comparable image fidelity.

1) Overcoming the Generative Learning Trilemma with LDDGAN

Deep generative models face a challenge known as the generative learning trilemma, which requires them to address three key requirements simultaneously: high sample quality, mode coverage, and fast sampling [28]. Although diffusion models have shown impressive sample quality and variety, their expensive sampling limits their practical application. DDGAN [28] and WDDGAN [29] have made significant progress in addressing the slow sampling weakness of diffusion models. The results in Table. 3 demonstrate that our model further improves upon this weakness, achieving state-of-the-art running speed among diffusion models while maintaining competitive sample quality. A few variants of diffusion models achieve better FID than our model, such as Score SDE [44] or DDPM [47]. However, our model achieves a sampling speed that is $5000 \times$ faster than Score SDE and $1000 \times$ faster than DDPM. In particular, compared to the two predecessors DDGAN and WDDGAN, our model outperforms on all metrics considered.

We plot the FID score against sampling time of various models in Figure. 3 to better benchmark our method. The figure clearly highlights the advantage of our model over

existing diffusion models. When comparing our model with GANs, only StyleGAN2 [60] with adaptive data augmentation has slightly better sample quality than our model. However, Table 3 shows that sample diversity is still their weakness, as their recall scores are below 0.5. Figure. 4 shows qualitative samples of CIFAR-10.

2) High-resolution image generation

To further evaluate the effectiveness of the model, we trained the model on two high-resolution datasets, LSUN Church and CELEBA-HQ. Table. 4 and 5 show the results of comparisons with previous models. Similar to CIFAR10, our model outperforms the two predecessors DDGAN and WDDGAN on all three comparison metrics. In particular, for the LSUN dataset, as shown in Table. 1, successfully compressing the input data by 8 times helps significantly reduce the inference time of the model. Our model is $1.5 \times$ faster than WDDGAN and $3 \times$ faster than DDGAN. When compared to other diffusion models such as DDPM or Score SDE, our model achieves comparable or better FID results. Compared to traditional GANs such as StyleGAN2, our model obtains competitive sample quality while outperforming GANs in terms of sample diversity. Figure. 5 shows qualitative samples of LSUN Church and CELEBA-HQ.

Model	FID↓	Recall↑	Time(s)↓
Ours	5.21	0.40	0.55 (*)
WDGAN [29]	5.94	0.37	0.60
DDGAN [28]	7.64	0.36	1.73
Score SDE [45]	7.23	-	-
NVAE [64]	29.7	-	-
VAEBM [65]	20.4	-	-
PGGAN [39]	8.03	-	-
VQ-GAN [35]	10.2	-	-

TABLE 4: Result on CELEBA-HQ 256×256 . (*) including reference time of the decoder.

Model	FID↓	Recall↑	Time(s)↓
Ours	4.67	0.42	1.02 (*)
WDGAN [29]	5.06	0.40	1.54
DDGAN [28]	5.25	0.41	3.42
DDPM [47]	7.89	-	-
ImageBART [66]	7.32	-	-
PGGAN [39]	6.42	-	-
StyleGAN [60]	4.21	-	-
StyleGAN2 [61]	3.86	0.36	-

TABLE 5: Result on LSUN Church 256×256 . (*) including reference time of the decoder.

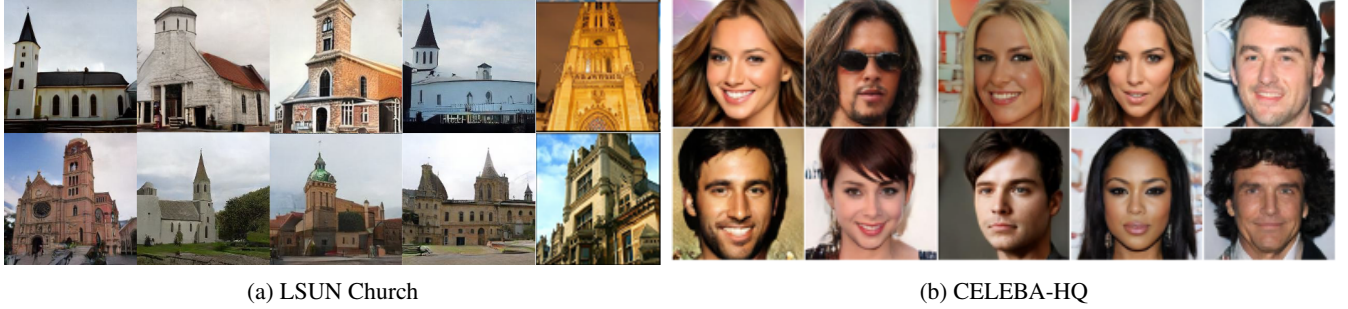


FIGURE 5: Qualitative results on LSUN Church and CELEBA-HQ 256×256 .

C. ABLATION STUDIES

1) Contribution of reconstruction loss and Weighted Learning

As described in Section. IV-C2, using reconstruction loss helps to achieve better FID but may reduce sample diversity. To take advantage of the effectiveness of reconstruction loss while maintaining high sample diversity, we proposed Weighted Learning, a new method of combining reconstruction loss and adversarial loss, instead of using a simple linear combination. In this section, we investigate the impact of reconstruction loss on the training results of the model as well as the contribution of Weighted Learning. We train our model with three different overall loss function for the generator: (i) adversarial loss alone, (ii) Simple linear combination of adversarial loss and reconstruction loss similar to Eq. 4, and (iii) combination of adversarial loss and reconstruction loss using Weighted Learning. Experimental results are summarized in Table. 6.

Dataset	Model	FID↓	Recall↑
CIFAR10	Ours (*)	3.15	0.58
	Ours + Recloss	3.09	0.56
	Ours + Recloss + WL	3.03	0.58
CELEBA HQ	Ours (*)	5.27	0.39
	Ours + Recloss	5.23	0.36
	Ours + Recloss + WL	5.21	0.40

TABLE 6: Contribution of reconstruction loss and Weighted Learning. "Recloss" denotes reconstruction loss, "WL" denotes Weighted Learning. (*) No reconstruction loss, only use adversarial loss.

As shown in Table. 6, employing reconstruction loss yields better FID scores than relying solely on adversarial loss for both datasets. This improvement arises because the reconstruction loss provides the generator with additional direct feedback regarding the disparity between the input training data and the data generated by the model, facilitating the rapid generation of data with superior image quality. However, reconstruction loss can also be interpreted as necessitating the generator to produce data identical to the input data using random noise. Consequently, after a certain amount of training, the generator tends to generate similar data with different input random noise, thereby reducing sample diversity. The

results in Table 6 illustrate this reduction in recall compared to scenarios without the use of reconstruction loss.

Weighted Learning is implemented to capitalize on reconstruction loss in the early stages, gradually diminishing its priority to emphasize adversarial loss and enhance sample diversity. Finally, the vanishing rate of reconstruction loss is reduced to ensure training stability. This strategic approach allows us to leverage reconstruction loss effectively, improving FID while ensuring sample diversity. As shown in Table. 6, the utilization of Weighted Learning helps us achieve better results for both FID and Recall.

2) Affect of learned latent space by Autoencoder

As outlined in Section. IV-B, introducing a KL-penalty towards a Gaussian distribution in the learned latent space proves effective for models relying on Gaussian-dependent structures ([30], [31], [32], [33]). Nevertheless, such a penalty is deemed unnecessary and potentially detrimental for our model, which operates with complex and multimodal distributions. To verify this hypothesis, we trained autoencoders with two settings: (i) using only perceptual loss and a patch-based adversarial loss as presented in Section. IV-B, (ii) using KL-penalty between the learned latent and a Gaussian distribution in addition to the two loss functions mentioned above. We selected autoencoders with reconstruction FID close to each other to compare the effectiveness objectively through FID, Recall of the main model. The results in Table. 7 show that when allowed to freely search for the appropriate latent space, autoencoders converges significantly faster, requiring fewer epochs to reach a reconstruction FID comparable to the case using KL-penalty. When comparing the results of FID and Recall of the main model, the ability to freely search for the appropriate latent space instead of relying on latent spaces close to a Gaussian distribution significantly improves the results in most cases. Especially in the case of CELEBA-HQ, even though using an autoencoder with a worse reconstruction FID than an autoencoder using KL-penalty, the main model still achieves better FID and Recall. FID is also significantly improved in the cases of CIFAR10 and LSUN. This result is a validation of our hypothesis.

	Autoencoder's FID ↓		LDDGAN's FID ↓		LDDGAN's Recall ↑	
	KL penalty	w/o KL penalty	KL penalty	w/o KL penalty	KL penalty	w/o KL penalty
CIFAR-10	1.33 (525ep)	1.32 (300ep)	3.15	3.03	0.58	0.58
CELEBA-HQ	0.56 (625ep)	0.58 (450ep)	5.88	5.83	0.37	0.40
LSUN Church	1.16 (675ep)	1.14 (525ep)	4.89	4.67	0.41	0.42

TABLE 7: Comparison results between using and not using KL-penalty between the learned latent and standard normal when training autoencoders. The autoencoders are selected to have FID reconstruction that are almost identical to each other in order to avoid affecting the objectivity of the comparison of the results of the main model. The numbers in (*) are the number of epochs needed to train the autoencoder.

3) The role of the autoencoder in achieving high performance

One evident observation from the experimental results is the strong dependence of our model on pre-trained autoencoders. A proficient autoencoder plays a crucial role in compressing images to their smallest form, thereby significantly reducing computational costs and enhancing inference speed. Moreover, this facilitates the training of the generator when the essential features have been effectively and comprehensively extracted by the encoder. The LSUN case in this study exemplifies this pattern, as demonstrated in Tables 2 and 5. Notably, well-designed autoencoders like this can also be repurposed for training future models with other tasks such as text2img or super-resolution.

Conversely, our model fails to achieve optimal performance without the presence of a proficient autoencoder, as evidenced by the results on CIFAR10 in Table 3. Despite compressing data only four times, which is not significantly different from WDDGAN [29], our model, leveraging latent space features, attains superior FID and Recall. However, the inference speed is almost equivalent to that of WDDGAN. Additionally, in this scenario, as shown in Table 2, the total number of parameters and FLOPs has increased compared to WDDGAN due to the inclusion of the autoencoder.

VI. CONCLUSION

In this paper, we present a new diffusion model called LDDGAN. The core concept of our approach involves transitioning from the complex, high-dimensional pixel space to an efficient, low-dimensional latent space using a pretrained autoencoder. Subsequently, the GAN mechanism facilitates the diffusion process, resulting in a notable reduction in computational costs, an acceleration in inference speed, and improvements in the quality and diversity of generated images. To further enhance the quality and diversity of images, we eliminate the dependence of latent learned by autoencoders on Gaussian distributions and propose Weighted Learning to combine reconstruction loss and adversarial loss. Our model achieves a state-of-the-art inference speed for a diffusion model. In comparison to GANs, we attain comparable image generation quality and inference speed, along with higher diversity. Furthermore, when compared to two predecessors DDGAN [28] and WDDGAN [29], our model outperforms them significantly across the majority of comparison metrics. With these initial results, we aim to foster future studies on

real-time and high-fidelity diffusion models.

APPENDIX A NETWORK CONFIGURATIONS

As mentioned in Sec. V-A4, we construct the generator based on NCSN++ [44] with configurations for each dataset as shown in Table 8. The same configuration is applied for both CELEBA-HQ and LSUN. For CIFAR10, due to the relatively small input size, the model is simplified to minimize computational costs. Regarding the discriminator, we design it based on ResBlock, which is used for the generator. Details of the discriminator's architecture are presented in Tab. 9. We use 3 ResBlocks for CIFAR10, 4 ResBlocks for CELEBA-HQ, and LSUN.

	CIFAR10	CELEBA-HQ	LSUN
# of ResNet blocks per scale	2	2	2
Base channels	128	128	128
Channel multiplier per scale	(1,2,2)	(1,2,2,2)	(1,2,2,2)
Attention resolutions	None	16	16
Latent Dimension	25	25	25
# of latent mapping layers	4	4	4
Latent embedding dimension	256	256	256

TABLE 8: Network configurations for the generator.

Discriminator
1 × 1 conv2d, 128
ResBlock, 128
ResBlock down, 256
ResBlock down, 512
ResBlock down, 512
minibatch std layer
Global Sum Pooling
FC layer → scalar

TABLE 9: Network structures for the discriminator. The number on the right is the number of channels of Conv in each residual block.

APPENDIX B TRAINING HYPER-PARAMETERS

Details of the hyperparameters used for the training process are presented in Table 9. It is noteworthy that we use a larger batch size than DDGAN and WDDGAN with fewer GPUs, thanks to the data compression capability of autoencoders. Regarding training times, CIFAR10 requires 1 day on 1 GPU, CELEBA-HQ requires 1.5 days on 2 GPUs, and LSUN requires 2.1 days on 2 GPUs.

	CIFAR10	CELEBA-HQ	LSUN
Learning rate of G	1.6e-4	2e-4	2e-4
Learning rate of D	1.25e-4	1e-4	1e-4
EMA	0.9999	0.999	0.999
Batch size	256	128	128
Lazy regularization	15	10	10
Number of epochs	1700	400	400
Number of timesteps	4	2	4
Number of GPUs	1	2	2

TABLE 10: Training hyper-parameters.

APPENDIX C CHOOSING BACKBONE FOR LATENT SPACE

When transitioning from pixel space to latent space, we carefully consider a suitable backbone for the generator in this space. We particularly pay attention to Vision Transformers (ViTs) models due to their ability to extract long-range relationships and high compatibility with latent space. Recently, a ViT-based Unet introduced by Xiao *et al.* in [68] has demonstrated the potential to replace the widely used CNN-based Unet in diffusion models. To compare the effectiveness of ViT and CNN backbones for tasks in this study, we construct another generator based on this model. Except for the generator's architecture, we maintain the core model as depicted in Figure 1. Timestep t and x_t are input to the generator as a token after passing through embedding layers. We introduce adaptive group normalization layers into the transformer blocks of the ViT-based Unet and control them with the latent variable z , similar to the CNN-based Unet described in Section. V. This helps the generator model denoising distributions within a complex and multimodal context. Other settings of the ViT-based Unet are kept consistent with [68]. We conduct experiments with three different scales of the model: Tiny, Mid, and Large, and subsequently compare the results with the CNN-based Unet.

Table. 11 presents the results of the computational cost comparison, and Table. 12 provides the comparison results on CIFAR10. It is evident that the ViT-based Unet achieves better results in both image quality and diversity than the CNN-based Unet. However, to outperform the CNN-based Unet, the ViT-based Unet requires more computational cost and inference time. Additionally, the CNN-based Unet converges faster, reaching its best results in only 1500 epochs, while the ViT-based Unet needs 2300 epochs. In a general comparison across five aspects: computational cost, convergence speed, inference speed, image quality, and diversity, the CNN-based Unet remains more suitable for our latent space. This result may be explained through the inherent inductive biases of CNN, which is also a weakness of ViT models. The inductive biases have helped the CNN-based Unet model features and distributions more rapidly, especially as we utilize two-dimensional latent variables, similar to the image data that CNNs are designed to process.

Model	Param (M) ↓	MEM (GB) ↓
CNN based	41.97	0.18
ViT based, Tiny	28.01	0.08
ViT based, Mid	43.32	0.21
ViT based, Large	56.17	0.32

TABLE 11: Comparison of computational cost between CNN-based and ViT-based Unet.

Model	FID ↓	Recall ↑	NFE ↓	Time(s) ↓
CNN based	3.03	0.58	4	0.08
ViT based, Tiny	4.33	0.54	4	0.06
ViT based, Mid	3.21	0.57	4	0.09
ViT based, Large	2.97	0.59	4	0.13

TABLE 12: Comparisonal results between CNN-based and ViT-based Unet on CIFAR-10.

APPENDIX D ADDITIONAL QUALITATIVE SAMPLES AND QUALITATIVE COMPARISON

We further provide additional qualitative samples and qualitative comparison on CelebA-HQ 256 in Figure. 6 and LSUN-Church in Figure. 7.

Our model clearly achieves better sample quality. On the CELEBA-HQ dataset, both DDGAN and WDDGAN face challenges in generating distinct and complete human faces, often producing distorted features. Similarly, on the LSUN Church dataset, these models struggle to accurately render straight and horizontal architectural details. Figure. 7a and 7b further illustrate these issues, showcasing images marred by numerous undesirable artifacts. Conversely, our model consistently produces images that are both realistic and sharply defined.

ACKNOWLEDGMENT

We would like to thank Hamagami Lab members (Department of Mathematics, Physics, Electrical Engineering, and Computer Science, Yokohama National University) for their very useful discussions and suggestions. Special thanks to Trang T.H. Nguyen for supporting the completion of this work.

REFERENCES

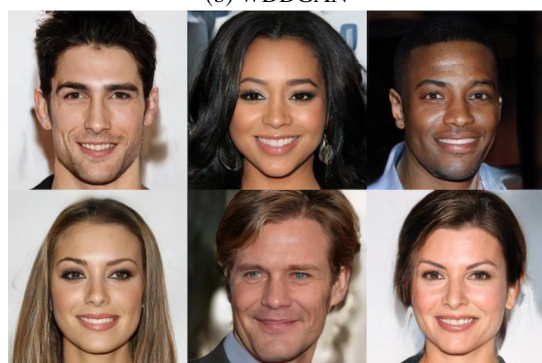
- [1] P. Dhariwal and A. Nichol. "Diffusion models beat gans on image synthesis." *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.
- [2] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes. "Photorealistic text-to-image diffusion models with deep language understanding." *arXiv preprint, arXiv:2205.11487*, 2022.
- [3] C. Huang, J. H. Lim, and A. C. Courville. "A variational perspective on diffusion-based generative models and score matching." *Advances in Neural Information Processing Systems*, 34:22863–22876, 2021.
- [4] D. Kingma, T. Salimans, B. Poole, and J. Ho. "Variational diffusion models." *Advances in neural information processing systems*, 34:21696–21707, 2021.
- [5] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. "Hierarchical text-conditional image generation with clip latents." *ArXiv preprint, arXiv:2204.06125*, 2022.
- [6] J. Choi, S. Kim, Y. Jeong, Y. Gwon, and S. Yoon. "Ilvr: Conditioning method for denoising diffusion probabilistic models". *ArXiv preprint, arXiv:2108.02938*, 2021.



(a) DDGAN

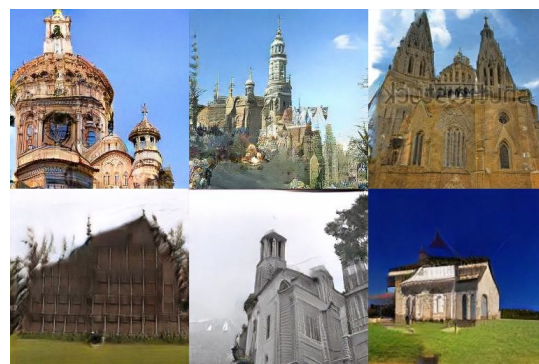


(b) WDDGAN



(c) Ours

FIGURE 6: Additional qualitative samples and qualitative comparison on CelebA-HQ 256×256 .



(a) DDGAN



(b) WDDGAN



(c) Ours

FIGURE 7: Additional qualitative samples and qualitative comparison on LSUN Church 256×256 .

- [7] C. Meng, Y. Song, J. Song, J. Wu, J. Zhu, and S. Ermon. "Sdedit: Image synthesis and editing with stochastic differential equations." ArXiv preprint, arXiv:2108.01073, 2021.
- [8] J. Ho, W. Chan, C. Saharia, J. Whang, R. Gao, A. Gritsenko, D. P. Kingma, B. Poole, M. Norouzi, D. J. Fleet. "Imagen video: High definition video generation with diffusion models." ArXiv preprint, arXiv:2210.02303, 2022.
- [9] M. Zhao, F. Bao, C. Li, and J. Zhu. Egsde: "Unpaired image-to-image translation via energy-guided stochastic differential equations." Computer Vision and Pattern Recognition, 2022.
- [10] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet. "Video diffusion models." Computer Vision and Pattern Recognition, 2022.
- [11] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall. "Dreamfusion: Text-to-3d using 2d diffusion." Computer Vision and Pattern Recognition, 2022.
- [12] I. J. Goodfellow, J. P. Abadie, M. Mirza, B. Xu, D. W. Farley, S. Ozair, A. C. Courville, and Y. Bengio. "Generative adversarial networks." CoRR, 2014.
- [13] J. D. P. Kingma and M. Welling. "Auto-Encoding Variational Bayes." In 2nd International Conference on Learning Representations, ICLR, 2014.
- [14] L. Dinh, D. Krueger, and Y. Bengio. "Nice: Non-linear independent components estimation", arXiv:1410.8516, 2015.
- [15] L. Dinh, J. Sohl-Dickstein, and S. Bengio. "Density estimation using real NVP." In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017.
- [16] M. Arjovsky, S. Chintala, and L. Bottou. "Wasserstein gan." arXiv:1701.07875, 2017.
- [17] A. Brock, J. Donahue, and K. Simonyan. "Large scale GAN training for high fidelity natural image synthesis." In Int. Conf. Learn. Represent., 2019.
- [18] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. "Improved training of wasserstein gans." arXiv:1704.00028, 2017.
- [19] L. M. Mescheder. "On the convergence properties of GAN training." CoRR, arXiv:1801.04406, 2018.
- [20] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. "Unrolled generative adversarial networks." In 5th International Conference on Learning Representations, ICLR 2017.
- [21] P. Dhariwal and A. Nichol. "Diffusion models beat gans on image synthe-

- sis." CoRR, arXiv:2105.05233, 2021.
- [22] R. San-Roman, E. Nachmani, and L. Wolf. "Noise estimation for generative diffusion models." arXiv preprint arXiv:2104.02600, 2021.
- [23] A. Jolicoeur-Martineau, K. Li, R. Piche-Taillefer, T. Kachman, and I. Mitliagkas. "Gotta go fast when generating data with score-based models." arXiv preprint arXiv:2105.14080, 2021a.
- [24] R. Abdal, Y. Qin, P. Wonka. "Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space?" Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 4432-4441, 2019.
- [25] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. "Deep unsupervised learning using nonequilibrium thermodynamics," In International Conference on Machine Learning, 2015
- [26] W. Feller. "On the theory of stochastic processes, with particular reference to applications", In Proceedings of the [First] Berkeley Symposium on Mathematical Statistics and Probability, pp. 403-432. University of California Press, 1949
- [27] F. Bao, C. Li, J. Zhu, and B. Zhang. "Analyticdpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models", International Conference on Learning Representations 2022, 2022
- [28] Z. Xiao, K. Keris, A. Vahdat. "Tackling The Generative Learning Trilemma With Denoising Diffusion Gans", International Conference on Learning Representations 2022
- [29] H. Phung, Q. Dao, A. Tran. "Wavelet Diffusion Models are fast and scalable Image Generators", International Conference on Learning Representations 2023
- [30] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models", arXiv:2112.10752v2, 2022.
- [31] D. P. Kingma and M. Welling. "Auto-Encoding Variational Bayes". In 2nd International Conference on Learning Representations, ICLR, 2014.
- [32] D. Jimenez Rezende, S. Mohamed, and D. Wierstra. "Stochastic backpropagation and approximate inference in deep generative models". In Proceedings of the 31st International Conference on International Conference on Machine Learning, ICML, 2014
- [33] O. Avarahami, O. Fried, D. Lischinski. "Blended Latent Diffusion", ACM Transactions on Graphics, Vol. 42, No. 4, Article 149.
- [34] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, F. Penna, R. Rombach. "SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis", arXiv:2307.01952, 2023.
- [35] P. Esser, R. Rombach, and B. Ommer. "Taming transformers for high-resolution image synthesis". CoRR, abs/2012.09841, 2020.
- [36] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. "The unreasonable effectiveness of deep features as a perceptual metric". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.
- [37] J. Yu, X. Li, J. Yu Koh, H. Zhang, R. Pang, J. Qin, A. Ku, Y. Xu, J. Baldridge, and Y. Wu. "Vector-quantized image modeling with improved vqgan", 2021.
- [38] A. Dosovitskiy and T. Brox. "Generating images with perceptual similarity metrics based on deep networks". In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, Adv. Neural Inform. Process. Syst., pages 658-666, 2016.
- [39] T. Karras, T. Aila, S. Laine, and J. Lehtinen. "Progressive growing of GANs for improved quality, stability, and variation". In International Conference on Learning Representations, 2018.
- [40] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao. "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop". arXiv preprint arXiv:1506.03365, 2015.
- [41] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. "Gans trained by a two time-scale update rule converge to a local nash equilibrium." Advances in neural information processing systems, 2017.
- [42] T. Kynkaanniemi, T. Karras, S. Laine, J. Lehtinen, and T. Aila. "Improved precision and recall metric for assessing generative models". arXiv preprint arXiv:1904.06991, 2019.
- [43] M. S. Sajjadi, O. Bachem, M. Lucic, O. Bousquet, and S. Gelly. "Assessing generative models via precision and recall". arXiv preprint arXiv:1806.00035, 2018.
- [44] Y. Song and S. Ermon. "Generative modeling by estimating gradients of the data distribution." In Advances in neural information processing systems, 2019.
- [45] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. "Score-based generative modeling through stochastic differential equations". In International Conference on Learning Representations, 2021c.
- [46] Y. Wu and K. He. "Group normalization." In Proceedings of the European conference on computer vision, 2018.
- [47] J. Song, C. Meng, and S. Ermon. "Denoising diffusion implicit models". In International Conference on Learning Representations, 2021a.
- [48] A. Jolicoeur-Martineau, R. P. Taillefer, I. Mitliagkas, and R. Tachet des Combes. "Adversarial score matching and improved sampling for image generation". In International Conference on Learning Representations, 2021b.
- [49] Y. Song, C. Durkan, I. Murray, and S. Ermon. "Maximum likelihood training of score-based diffusion models." arXiv preprint arXiv:2101.09258, 2021b.
- [50] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. "Score-based generative modeling through stochastic differential equations." In International Conference on Learning Representations, 2021c.
- [51] A. Vahdat, K. Kreis, and J. Kautz. "Score-based generative modeling in latent space." In Advances in neural information processing systems, 2021.
- [52] J. Song, C. Meng, and S. Ermon. "Denoising diffusion implicit models." In International Conference on Learning Representations, 2021a.
- [53] T.M. Cover and J. A. Thomas. "Elements of information theory". Wiley, New York, 1991.
- [54] J. Shlens. "Notes on Kullback-Leibler Divergence and Likelihood." arXiv:1404.2000, 2014
- [55] Z. Kong and W. Ping. "On fast sampling of diffusion probabilistic models." arXiv preprint arXiv:2106.00132, 2021.
- [56] R. Gao, Y. Song, B. Poole, Y. Nian Wu, and D. P. Kingma. "Learning energy-based models by diffusion recovery likelihood." In International Conference on Learning Representations, 2021.
- [57] E. Luhman and T. Luhman. "Knowledge distillation in iterative generative models for improved sampling speed." arXiv preprint arXiv:2101.02388, 2021.
- [58] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. "Spectral normalization for generative adversarial networks." arXiv preprint arXiv:1802.05957, 2018.
- [59] X. Gong, S. Chang, Y. Jiang, and Z. Wang. "Autogan: Neural architecture search for generative adversarial networks." In Proceedings of the IEEE conference on computer vision and pattern recognition, 2019.
- [60] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila. "Training generative adversarial networks with limited data." In Advances in neural information processing systems, 2020a.
- [61] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. "Analyzing and improving the image quality of stylegan." In Proceedings of the IEEE conference on computer vision and pattern recognition, 2020b.
- [62] D. P. Kingma and P. Dhariwal. "Glow: Generative flow with invertible 1x1 convolutions." In Advances in neural information processing systems, 2018.
- [63] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu. "Pixel recurrent neural networks." International Conference on Machine Learning, 2016b.
- [64] A. Vahdat and J. Kautz. "NVAE: A deep hierarchical variational autoencoder." In Advances in neural information processing systems, 2020.
- [65] Z. Xiao, K. Kreis, J. Kautz, and A. Vahdat. "Vaebm: A symbiosis between variational autoencoders and energy-based models." In International Conference on Learning Representations, 2021.
- [66] P. Esser, R. Rombach, A. Blattmann, and B. Ommer. "Imagebart: Bidirectional context with multinomial diffusion for autoregressive image synthesis." Advances in Neural Information Processing Systems, 34:3518-3532, 2021.
- [67] I. Anokhin, K. Demochkin, T. Khakhulin, G. Sterkin, V. Lempitsky, and D. Korzhnikov. "Image generators with conditionally-independent pixel synthesis." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021.
- [68] F. Bao, S. Nie, K. Xue, Y. Cao, C. Li, H. Su, J. Zhu. "All are Worth Words: A ViT Backbone for Diffusion Models." Computer Vision and Pattern Recognition, 2023.
- [69] T. Yin, M. Gharbi, R. Zhang, E. Shechtman, F. Durand, W. T. Freeman, T. Park. "One-step Diffusion with Distribution Matching Distillation", arXiv:2311.18828, 2023.
- [70] A. Sauer, D. Lorenz, A. Blattmann, R. Rombach, "Adversarial Diffusion Distillation", arXiv:2311.17042, 2023.



LUAN THANH TRINH received the B.S. and M.S. degrees in artificial intelligence from Yokohama National University, Japan, in 2021 and 2024 respectively.

His research interests include generative artificial intelligence, reinforcement learning, and explainable artificial intelligence.



TOMOKI HAMAGAMI Professor Tomoki Hamagami received his B.S. in Electrical Engineering, and Ph.D. in 1988, 1999 respectively from Chiba University. From 1988 to 2000, he was working on Intelligent System Laboratories in SECOM Co.,Ltd. as a senior researcher. From 2001 to 2004, he has worked in the Graduate school of Engineering, Chiba University as an assistant professor. In 2001, he moved to Yokohama National University as an associate professor in 2004, where he was

made a professor in 2008. His research interests include artificial intelligence, machine learning, reinforcement learning, multiagent systems, and smart systems with IoT and CPS.

...