DC-AE 1.5: Accelerating Diffusion Model Convergence with Structured Latent Space

Junyu Chen[†] Dongyun Zou Wenkun He Junsong Chen Enze Xie Song Han Han Cai[†]
NVIDIA
https://github.com/dc-ai-projects/DC-Gen

Abstract

We present DC-AE 1.5, a new family of deep compression autoencoders for high-resolution diffusion models. Increasing the autoencoder's latent channel number is a highly effective approach for improving its reconstruction quality. However, it results in slow convergence for diffusion models, leading to poorer generation quality despite better reconstruction quality. This issue limits the quality upper bound of latent diffusion models and hinders the employment of autoencoders with higher spatial compression ratios. We introduce two key innovations to address this challenge: i) ***Structured Latent Space****, a training-based approach to impose a desired channel-wise structure on the latent space with front latent channels capturing object structures and latter latent channels capturing image details; ii)* ***Augmented Diffusion Training****, an augmented diffusion training strategy with additional diffusion training objectives on object latent channels to accelerate convergence. With these techniques, DC-AE 1.5 delivers faster convergence and better diffusion scaling results than DC-AE. On ImageNet 512×512, DC-AE-1.5-f64c128 delivers better image generation quality than DC-AE-f32c32 while being 4× faster.*

1. Introduction

Latent Diffusion Model (LDM) [25] has established itself as a dominant paradigm in high-resolution image synthesis [12, 19, 21, 32]. It leverages an autoencoder to project images of size $H \times W \times 3$ into compressed latent representations of shape $\frac{H}{f} \times \frac{W}{f} \times c$[1], reducing the computational cost of the diffusion model. The diffusion model's outputs are fed to the autoencoder at inference time to reconstruct images from the latent representations. In LDM, the autoencoder reconstruction quality is critical, as it sets a quality upper bound for the whole image synthesis pipeline.

A common approach to improve the quality of autoencoder reconstruction is to increase its latent channel number [10, 12, 19]. For example, Figure 1 (b) shows the autoencoder's rFID (reconstruction FID, the lower the better) results with different latent channel numbers. We can see that the rFID consistently improves as the latent channel number increases, decreasing from 1.60 to 0.26 if switching from c16 to c128.

Expanding the latent channel number is especially critical for deep compression autoencoders [7], which accelerate latent diffusion models by increasing the autoencoder's spatial compression ratio (Figure 1a). Under a high spatial compression ratio (e.g., f64), deep compression autoencoders must use a large latent channel number (e.g., c128) to maintain a satisfactory reconstruction quality.

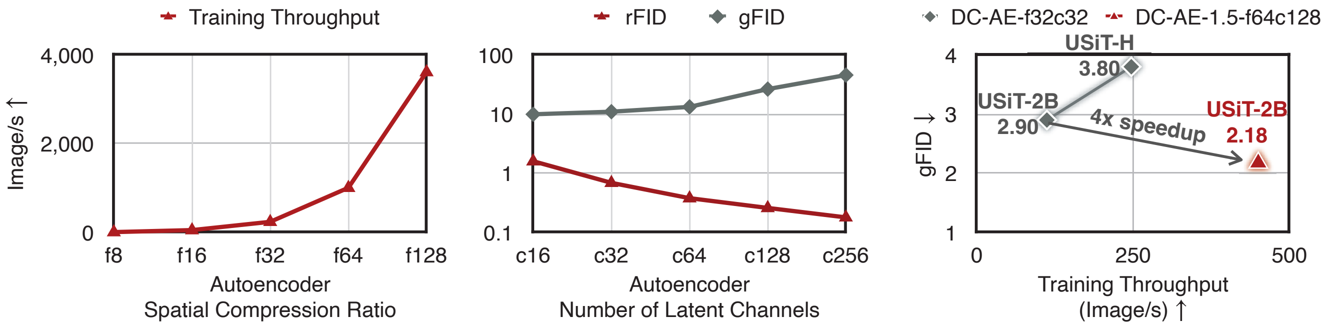
However, using a large latent channel number significantly slows the diffusion model's convergence, leading to worse gFID (generation FID, the lower the better) results. For example, Figure 1 (b) demonstrates DiT-XL's [24] gFID results with different latent channel numbers. Although the autoencoder's rFID keeps improving, the gFID keeps deteriorating. This issue not only limits LDM's quality upper bound but also bounds its efficiency as it hinders the employment of autoencoders with high spatial compression ratios (e.g., f64).

This work presents DC-AE 1.5, which introduces two key innovations to tackle the aforementioned challenge. First, we analyze the autoencoder's latent space under different latent channel numbers (Section 3.1). We find that the latent space has a sparsity issue when using a large number of latent channels (Figure 2a). It allocates most of the latent channels for capturing image details while the latent channels for capturing object structure become sparse among the whole latent space[2]. This sparsity issue makes it more difficult for diffusion models to learn object structure, leading to the slow convergence issue. As shown in Figure 2 (b), the image details remain good when we increase the latent

[†]Equal contribution. Correspondence to: Han Cai (hcai@nvidia.com).

[1]'f' denotes the autoencoder's spatial compression ratio. 'c' represents the latent channel number of the autoencoder.

[2]We refer to them as *detail latent channels* and *object latent channels*, respectively.



(a) (b) (c)

Figure 1. **(a) Training Throughput Comparison under Different Autoencoder Spatial Compression Ratios.** Increasing the autoencoder's spatial compression ratio effectively improves diffusion models' training efficiency by producing a latent space with fewer tokens. However, larger latent channel numbers are required to maintain satisfactory reconstruction quality. **(b) rFID and gFID Results under Different Latent Channel Numbers.** We use DiT-XL [24] as the diffusion model. rFID keeps improving with more latent channels, while gFID keeps getting worse. **(c) Efficiency-Quality Trade-off Comparison on ImageNet 512×512.** Classifer-free guidance [13] is not used. DC-AE-1.5-f64c128 delivers 4× speedup over DC-AE-f32c32 while maintaining a better image generation quality.

channel number. However, the object structures suffer from significant distortion.

Motivated by the findings, we propose **Structured Latent Space** (Section 3.2) to alleviate the sparsity issue. It introduces a training-based approach (Figure 4) to imposing a specific structure on the latent space: the front latent channels focus on capturing object structure, while the latter latent channels focus on capturing image details (Figure 3). Second, based on the Structured Latent Space, we propose **Augmented Diffusion Training** (Section 3.3) to address the slow convergence issue. It introduces extra diffusion training objectives on object latent channels to accelerate diffusion models' learning speed on capturing object structure (Figure 5).

With these techniques, we significantly accelerate diffusion models' convergence when using a large latent channel number (Figure 5b), leading to better gFID results (Table 2 and 3). In addition, it also leads to better diffusion model scaling results than previous autoencoders (e.g., DC-AE-f32c32), as shown in Figure 6. On ImageNet 512×512, DC-AE-1.5-f64c128+USiT-2B achieves 2.18 gFID without classifier-free guidance, surpassing DC-AE-f32c32+USiT-2B while being 4× faster (Figure 1c). We summarize our contributions as follows:

- We analyze the latent space of autoencoders under different latent channel numbers. Our study reveals the latent space's sparsity issue when using a large latent channel number, which leads to the slow convergence issue for diffusion models.
- We propose Structured Latent Space and Augmented Diffusion Training, which effectively address the slow convergence issue.
- We build DC-AE 1.5 based on our key innovations. It delivers better diffusion scaling results than prior autoencoders, boosting the quality upper bound of LDMs and paving the way for the employment of autoencoders with higher spatial compression ratios.

2. Related Work

Autoencoders in Image Generation. Training diffusion models in high-resolution pixel space is computationally prohibitive due to the large token number. To overcome this challenge, [25] proposes to employ pretrained autoencoders to produce a compressed latent space and operate diffusion models on the compressed space. This strategy significantly reduces diffusion models' training and inference costs, making it possible to train large-scale diffusion models on web-scale datasets, leading to impressive image generation results [12, 19, 21].

Following that, one line of works focuses on improving the autoencoders' reconstruction quality by increasing the latent channel number [10, 12, 19] or adding task-specific priors [37]. Another line of works focuses on improving diffusion models' efficiency by increasing autoencoders' spatial compression ratio [5–7]. A third line of work [9, 27, 29, 30] explores hierarchical VAEs with multi-resolution latent spaces, while our approach focuses on introducing structure in the latent channel dimension. Concurrent to our work, some recent works propose to make the latent space more friendly for diffusion models by aligning it with a pretrained discriminative model [33] or leveraging scale equivariance regularization [16, 28]. Unlike prior works, our work addresses the slow convergence issue of diffusion models when using a large latent channel number (e.g., c128). It also provides novel insights and solutions to this critical problem.

Accelerating Diffusion Model Convergence. Training the diffusion model consumes extensive computation resources, motivating many works to improve its training

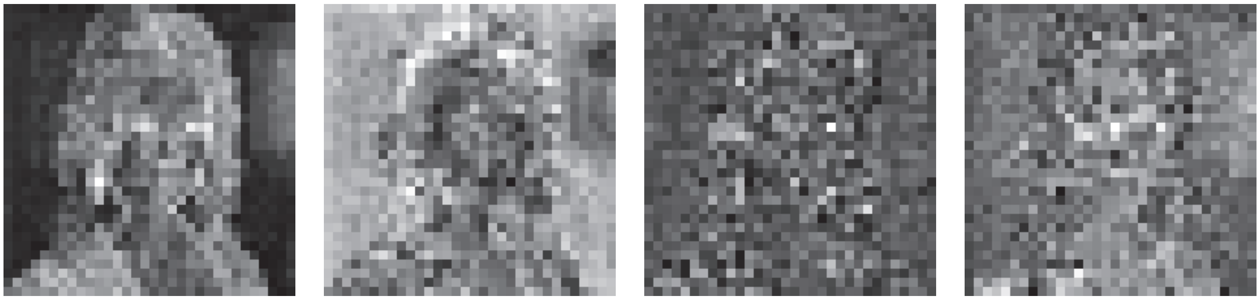


Figure 2. We visualize the channel-wise average feature here. We provide the complete latent space visualization in the supplementary material (Figure 9 and 10). The visualization shows that the object structure information gets blurred if we increase the latent channel number. It makes diffusion models unable to learn object structure efficiently. As a result, we can see gradually distorted object structures when we enlarge the latent channel number, as shown in the visualization of the diffusion model's outputs. We use the DiT-XL as the diffusion model here.

efficiency. Representative techniques include designing better model architectures [2–4], representation alignment with pretrained discriminative models [35], better diffusion training schedulers [12, 22], improving the training data quality [8, 31], etc. Orthogonal to these techniques, our work accelerates diffusion model convergence by employing a structured latent space.

3. Method

In Section 3.1, we first analyze the latent spaces of autoencoders with different latent channel numbers to understand the underlying source of the slow convergence issue. Next, in Section 3.2 and Section 3.3, we introduce DC-AE 1.5 with *Structured Latent Space* and *Augmented Diffusion Training* to address the slow convergence issue.

3.1. Analysis and Motivation

We visualize the latent spaces of DC-AE-f32 [7] with different latent channels (c32, c64, c128, and c256) to analyze why diffusion models suffer from the slow convergence issue when using a large latent channel number. Due to the page limit, we provide the complete latent space visualization in the supplementary material. In Figure 2 (a), we provide the visualization of the channel-wise average feature of the latent spaces[3].

We can see that the information about the object structure becomes more sparse as we increase the latent channel number from c32 to c256. From the complete latent space visualization (Figure 9), we find that this phenomenon is because the f32c256 autoencoder contains a lot of latent channels for capturing image details. In contrast, only a few latent channels are responsible for capturing object structure.

It is reasonable since capturing more image details is critical for achieving a good reconstruction quality. However, since we treat all latent channels equally, it is challenging for diffusion models to distinguish between the overall object structure and high-frequency details, making them unable to learn object structure efficiently. For example, we visualize the image generation results of DiT-XL [24] with different autoencoders (DC-AE-f32c32 $\rightarrow$ DC-AE-f32c256) in Figure 2 (b). We can see that the diffusion model gradually loses control over structural coherence as the number of latent channels increases. In contrast, the image details remain good under all settings.

[3]Assuming each latent representation has a shape of $[h, w, c]$, we get its channel-wise average feature via 'torch.mean(latent, dim=2)'.

Figure 3. **Image Reconstruction Comparison.** With the structured latent space, DC-AE 1.5 can reconstruct images given partial latent channels, with front latent channels reconstructing overall object structure and semantics and latter latent channels adding details. In contrast, DC-AE can not reconstruct the object structure well given partial latent channels. The decoder is fine-tuned for all settings to fully reveal the information encoded in the (partial) latent space.

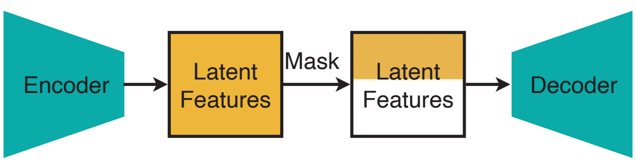


Figure 4. **Illustration of DC-AE 1.5 Autoencoder Training Strategy.** The key difference from conventional autoencoder training is that we add a channel-wise random masking step before feeding the latent features to the decoder. The mask is generated randomly at each step according to Eq. 1. It enables the autoencoder to reconstruct with partial latent channels and naturally impose the channel-wise structure on the latent space.

Based on these findings, we conjecture *the autoencoder's latent space suffers from the object information sparsity issue as demonstrated in Figure 2 (a) when using a large latent channel number. This sparsity issue makes diffusion models unable to learn object structure efficiently, leading to slow convergence.*

3.2. Structured Latent Space

Motivated by the above analysis, we introduce Structured Latent Space to help diffusion models distinguish object latent channels from detail latent channels to tackle the latent space sparsity issue. Figure 3 illustrates the general idea. Unlike the conventional autoencoders, whose latent spaces have no structure along the latent channel dimension, we design and add a channel-wise structure on the latent space.

Specifically, DC-AE 1.5 has an additional capacity of reconstructing images from partial latent channels (e.g., first 16/32/64 channels from c128). For example, as shown in Figure 3, it first focuses on reconstructing object structures from the first 16 latent channels and gradually refines image details as we include more latent channels. In contrast, when given partial latent channels, the conventional autoencoder (e.g., DC-AE [7]) cannot reconstruct the object structure well.

In the supplementary material, we visualize the complete latent space comparison between DC-AE and DC-AE 1.5 (Figure 10). We can see a clear separation between object latent channels and detail latent channels in DC-AE 1.5's latent space with front channels as object latent channels and latter channels as detail latent channels, which is absent in DC-AE's latent space.

Figure 4 demonstrates our autoencoder training strategy to achieve the desired channel-wise latent space structure. The intuition of our design is based on the finding that the autoencoder's latent space naturally focuses more on the object structure when the latent channel number is small. Therefore, we augment the original autoencoder training objective with the additional objective of reconstructing input images from partial latent channels.

Specifically, an autoencoder consists of an encoder E that maps the input image $x \in \mathbb{R}^{H\times W\times 3}$ to the latent fea-

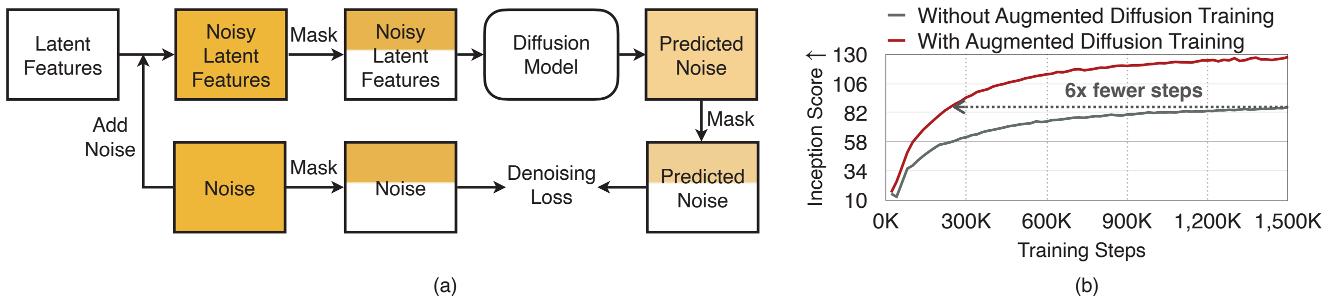


Figure 5. **(a) Illustration of Augmented Diffusion Training.** We randomly generate a channel-wise mask at each training step and use it to augment diffusion training. **(b) Training Curve Comparison.** We achieve 6× faster convergence on UViT-H [2] with Augmented Diffusion Training.

Autoencoder	Latent Shape	Structured Latent Space	rFID ↓
DC-AE-f32c32 [7]	8×8×32	✗	0.69
DC-AE-f32c128 [7]	8×8×128	✗	0.26
DC-AE-1.5-f32c128	8×8×128	✓	0.26

Table 1. **Image Reconstruction Results on ImageNet 256×256.** DC-AE 1.5 maintains the same rFID as DC-AE under the same setting (f32c128). In addition, DC-AE 1.5 has the latent space structure, while DC-AE does not.

ture $z = E(x) \in \mathbb{R}^{\frac{H}{f} \times \frac{W}{f} \times c}$ and a decoder D that predicts the image $y = D(z) \in \mathbb{R}^{H \times W \times 3}$ from the latent feature. A training loss $l(x, y) = l(x, D(z))$ is used to supervise the autoencoder's training, which is a weighted average of the L1 loss, the perceptual loss [36], and the GAN loss [14].

In our design, we also require the autoencoder to reconstruct the image when only the first several channels are selected. In practice, we implement this by randomly sampling a latent channel number ($c' \le c, c' \in [c_1, c_2, \cdots, c]$) at each training step and generate a mask

$$\text{mask}_{c,c'} = (\underbrace{1, 1, \ldots, 1}_{c'}, \underbrace{0, 0, \ldots, 0}_{c-c'}). \tag{1}$$

Then we use the modified loss $l(x, D(z \cdot \text{mask}_{c,c'}))$ to update the autoencoder.

Through this training strategy, the autoencoder gains the ability to reconstruct images given any latent channel number $c' \in [c_1, c_2, \cdots, c]$, and its latent space naturally has our desired channel-wise structure. In addition, this training strategy has little impact on the autoencoder's reconstruction quality. Table 1 shows the rFID comparison between DC-AE and DC-AE 1.5. DC-AE 1.5 achieves the same rFID as DC-AE under the same setting while having the additional latent space structure.

3.3. Augmented Diffusion Training

Given the structured latent space, our next step is to utilize this structure to accelerate diffusion models' learning efficiency on object structure for better convergence. We achieve this with Augmented Diffusion Training, as demonstrated in Figure 5 (a). The core idea is explicitly augmenting the diffusion training with additional objectives on object latent channels.

For example, let's consider a latent diffusion model ϵ_θ supervised by the noise prediction loss. Given the latent feature $x_0 \in \mathbb{R}^{\frac{H}{f} \times \frac{W}{f} \times c}$ and the corresponding noisy latent feature $x_t = \alpha_t x_0 + \beta_t \epsilon$, the denoising loss is defined as $\|\epsilon - \epsilon_\theta(x_t, t)\|^2$.

At each diffusion training step, we randomly sample a latent channel number ($c' \le c, c' \in [c_1, c_2, \cdots, c]$), and leverage the corresponding mask $\text{mask}_{c,c'}$ defined in Eq. 1 to modify the diffusion training loss

$$\|\epsilon \cdot \text{mask}_{c,c'} - \epsilon_\theta(x_t \cdot \text{mask}_{c,c'}, t) \cdot \text{mask}_{c,c'}\|^2. \tag{2}$$

Figure 5 (b) illustrates the training curves of UViT-H [2] with and without Augmented Diffusion Training on ImageNet 256×256, demonstrating 6× faster convergence and better final image generation quality.

4. Experiments

4.1. Setups

We use Pytorch [1] to implement and train our models on NVIDIA H100 GPUs. We use FSDP and bf16 training to reduce the training time and training memory cost. Our largest model (USiT-3B) takes around 5 days to complete training on 16 NVIDIA H100 GPUs.

For autoencoder training, we follow the same training strategy proposed in DC-AE [7], including a low-resolution full training phase, a high-resolution latent adaptation phase, and a local refinement phase. We add the training technique discussed in Section 3.2 to add the channel-wise structure to the latent space. Specifically, our autoen-

ImageNet 256×256				
Diffusion Model	Autoencoder	Patch Size	gFID ↓	Inception Score ↑
DiT-XL [24]	DC-AE-f32c128 [7]	1	26.44	53.41
	DC-AE-1.5-f32c128	1	**17.31**	**80.38**
UViT-H [2]	DC-AE-f32c128 [7]	1	17.38	78.42
	DC-AE-1.5-f32c128	1	**10.82**	**109.23**
SiT-XL [22]	DC-AE-f32c128 [7]	1	20.81	68.69
	DC-AE-1.5-f32c128	1	**14.91**	**93.78**
USiT-H [7]	DC-AE-f32c128 [7]	1	8.45	121.09
	DC-AE-1.5-f32c128	1	**6.10**	**146.03**

Table 2. **Comparison with DC-AE on ImageNet 256×256 Image Generation.** Faster convergence leads to better final results. DC-AE-1.5-f32c128 outperforms DC-AE-f32c128 under all settings.

coders support reconstructing with any latent channel number in $[16, 20, 24, ..., c]$, where c is the largest latent channel number (e.g., 128).

We follow the same training settings as the official implementations for diffusion experiments, except that we increase the training batch size from 256 to 1024 for all DiT and SiT models to increase GPU utilization. We consider 4 settings in our experiments, including DiT [24], UViT [2], SiT [22], and USiT [7]. In addition to existing diffusion models, we build a larger version of USiT to conduct diffusion model scaling experiments, with a depth of 56, a hidden dimension of 2048, and 32 heads. We refer to this model as USiT-3B. By default, we do not use classifier-free guidance [13] except stated explicitly. Evaluation metrics include FID [23], Inception Score [26], Precision [17], and Recall [17]. We profile the training throughputs of diffusion models on the NVIDIA H100 GPU with PyTorch.

4.2. Main Results

Accelerating Diffusion Model Convergence. We compare DC-AE 1.5 with DC-AE [7] under the same settings (f32c128 for ImageNet 256×256 and f64c128 for ImageNet 512×512) to evaluate DC-AE 1.5's effectiveness.

Table 2 summarizes the class-conditional image generation results on ImageNet 256×256 [11]. DC-AE-1.5-f32c128 not only converges faster than DC-AE-f32c128 but also results in better image generation quality, thanks to the improved convergence. It consistently delivers better gFID and Inception Scores than DC-AE-f32c128 under all settings. For example, on UViT-H [2], it improves gFID from 17.38 to 10.82 and Inception Score from 78.42 to 109.23.

Table 3 reports the class-conditional image generation results on ImageNet 512×512. The results are consistent with our findings on ImageNet 256×256. DC-AE-1.5-f64c128 outperforms DC-AE-f64c128 for all cases. It again

ImageNet 512×512				
Diffusion Model	Autoencoder	Patch Size	gFID ↓	Inception Score ↑
DiT-XL [24]	DC-AE-f64c128 [7]	1	21.31	71.17
	DC-AE-1.5-f64c128	1	**15.16**	**95.69**
UViT-H [2]	DC-AE-f64c128 [7]	1	17.34	88.47
	DC-AE-1.5-f64c128	1	**10.32**	**121.08**
SiT-XL [22]	DC-AE-f64c128 [7]	1	16.83	89.17
	DC-AE-1.5-f64c128	1	**12.44**	**114.08**
USiT-H [7]	DC-AE-f64c128 [7]	1	6.88	144.24
	DC-AE-1.5-f64c128	1	**5.35**	**163.03**

Table 3. **Comparison with DC-AE on ImageNet 512×512 Image Generation.** DC-AE-1.5-f64c128 outperforms DC-AE-f64c128 under all settings.

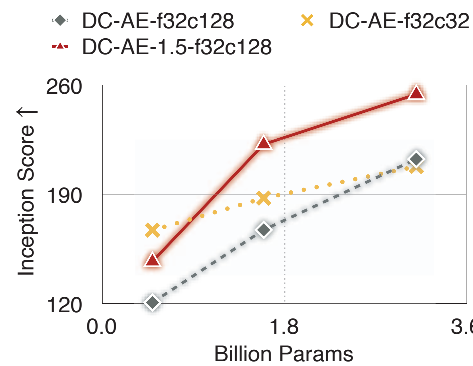


Figure 6. **Diffusion Model Scaling Results on ImageNet 256×256 with USiT.** DC-AE-1.5-f32c128 delivers a better scaling curve than DC-AE-f32c32 and DC-AE-f32c128.

justifies DC-AE 1.5's effectiveness in accelerating diffusion model convergence and improving image generation quality for autoencoder settings where we target using a large latent channel number (e.g., c128).

Improving Diffusion Model Scaling Curve. With DC-AE 1.5 addressing the slow convergence issue, an exciting thing is to boost the quality upper bound of latent diffusion models by using an autoencoder with a higher latent channel number to achieve a better image reconstruction quality. To demonstrate this, we conduct diffusion model scaling experiments with USiT [7] on ImageNet 256×256 and summarize the results in Figure 6.

Using DC-AE-f32c32 [7], we can see that the inception score improvements from scaling up the diffusion models saturate due to the autoencoder's limited reconstruction quality. In contrast, scaling up diffusion models with DC-AE-f32c128 brings more significant improvements. However, due to the slow convergence, its generation quality remains inferior to DC-AE-f32c32 until the model is scaled to USiT-3B. Last, using DC-AE-1.5-f32c128, we improve

ImageNet 512×512									
Image Generative Model	Autoencoder	Patch Size	Params (B)	Training Throughput (image/s) ↑	gFID ↓ w/o CFG	gFID ↓ w/ CFG	Inception Score ↑	Precision ↑	Recall ↑
UViT-H [2]	Flux-VAE-f8c16 [19]	2	0.50	55	30.91	12.63	56.72	0.64	0.59
DiT-XL [24]	Flux-VAE-f8c16 [19]	2	0.68	54	27.35	8.72	53.09	0.68	0.61
DiT-XL [24]	SD-VAE-f8c4 [25]	2	0.67	54	12.03	3.04	105.25	0.75	0.64
UViT-H [2]	SD-VAE-f8c4 [25]	2	0.50	55	11.04	3.55	125.08	0.75	0.61
SiT-XL [22]	SD-VAE-f8c4 [25]	2	0.67	54	-	2.62	-	-	-
MAGVIT-v2 [34]	-	-	-	-	3.07	1.91	213.1	-	-
MAR-L [20]	-	-	-	-	2.74	1.73	205.2	-	-
EDM2-XXL [15]	-	-	-	-	1.91	1.81	-	-	-
SiT-XL [22]	DC-AE-f32c32 [7]	1	0.67	241	7.47	2.41	131.37	0.77	0.65
USiT-H [7]	DC-AE-f32c32 [7]	1	0.50	247	3.80	1.89	174.58	0.78	0.64
USiT-2B [7]	DC-AE-f32c32 [7]	1	1.58	112	2.90	1.72	187.68	0.79	0.63
USiT-H [7]	DC-AE-f64c128 [7]	1	0.50	984	6.88	2.90	144.24	0.77	0.62
USiT-2B [7]	DC-AE-f64c128 [7]	1	1.58	450 4× faster	3.20 -0.72	2.33	195.94	0.80	0.61
USiT-2B [7]	DC-AE-1.5-f64c128	1	1.58	450	2.18	1.84	237.11	0.80	0.62
USiT-3B [7]	DC-AE-1.5-f64c128	1	3.11	214	1.70	1.63†	262.04	0.80	0.61

Table 4. **Comparison with State-of-the-Art Image Generative Models on ImageNet 512×512 Class-Conditional Image Generation.** DC-AE-1.5-f64c128 outperforms DC-AE-f32c32, delivering 4× higher training throughput and better image generation quality. † denotes the result obtained by using Guidance Interval [18].

the scaling curve by accelerating convergence. DC-AE-1.5-f32c128 can achieve superior Inception Score than DC-AE-f32c32 on USiT-2B and USiT-3B.

Comparison with State-of-the-Art Image Generative Models on ImageNet 512×512. In addition to improving the quality upper bound of latent diffusion models, another exciting direction is to push their efficiency frontier by increasing the spatial compression ratio, such as from f32c32 to f64c128. DC-AE 1.5 contributes to this direction by improving f64c128's image generation results with faster convergence. To demonstrate this, we train the USiT model (USiT-2B) with DC-AE-1.5-f64c128 on ImageNet 512×512 and compare our results with state-of-the-art diffusion models and autoregressive image generative models.

We summarize the results in Table 4. With DC-AE-1.5-f64c128, USiT-2B achieves a competitive image generation quality with exceptional efficiency. In particular, it achieves a gFID of 2.18, an inception score of 237.11 without using classifier-free guidance, significantly outperforming DC-AE-f32c32+USiT-2B. More importantly, it delivers 4× higher training throughput than DC-AE-f32c32+USiT-2B. Figure 7 demonstrates our image generation examples, showing competitive visual quality compared with other generative models trained on ImageNet.

4.3. Ablation Study

Component-Wise Analysis of DC-AE 1.5. We conduct component-wise ablation to investigate the impact of DC-AE 1.5's two key designs, i.e., Structured Latent Space (Section 3.2) and Augmented Diffusion Training (Section 3.3). If Structured Latent Space is enabled, we employ DC-AE-1.5-f32c128; otherwise, we use DC-AE-f32c128.

ImageNet 256×256				
Diffusion Model	Structured Latent Space	Augmented Diffusion Training	gFID ↓	Inception Score ↑
DiT-XL [24]			26.44	53.41
	✓		26.75	51.07
		✓	36.83	42.22
	✓	✓	**17.31**	**80.38**
USiT-H [7]			8.45	121.09
	✓		6.72	132.02
		✓	8.99	125.89
	✓	✓	**6.10**	**146.03**

Table 5. **Component-wise Ablation Study Experiments.** Removing either Structured Latent Space or Augmented Diffusion Training causes a significant image generation quality drop, showing it is critical to use these two techniques together.

We report the results in Table 5. We can see that both Structured Latent Space and Augmented Diffusion Training are essential for improving the image generation quality. For example, on DiT-XL [24], adding Structured Latent Space or Augmented Diffusion Training alone even hurts the results. In contrast, combining them leads to clear improvements, decreasing the gFID from 26.44 to 17.31 and increasing the inception score from 53.41 to 80.38. Thus, we recommend using these two techniques together.

Figure 7. **Selected Image Samples Generated by Diffusion Models using DC-AE 1.5.**

ImageNet 256×256				
Diffusion Model	Setting	Autoencoder	gFID ↓	Inception Score ↑
DiT-XL [24]	f32c32	DC-AE [7]	10.18	107.49
		DC-AE-1.5	10.50	107.99
USiT-H [7]	f32c32	DC-AE [7]	3.89	167.40
		DC-AE-1.5	4.15	166.94

Table 6. **Comparison with DC-AE under f32c32.**

Comparison with DC-AE under f32c32. While DC-AE 1.5 is mainly for autoencoder settings with large latent channel numbers (e.g., c128), we can still apply it to settings with small latent channel numbers (e.g., c32). Table 6 summarizes the comparison between DC-AE 1.5 and DC-AE under f32c32. We find DC-AE-f32c32 performs slightly better than DC-AE-1.5-f32c32. We conjecture that f32c32 does not have the latent space sparsity issue. Thus, adding Structured Latent Space and Augmented Diffusion Training is not necessary. It is consistent with our analysis in Section 3.1 and Figure 2. Based on this finding, we recommend using DC-AE 1.5 when targeting a large latent channel number (e.g., c128) and conventional autoencoders such as DC-AE when targeting a small latent channel number (e.g., c32).

5. Conclusion

This paper presents a new approach for addressing the slow convergence issue of latent diffusion models when using a large latent channel number. Our approach consists of two key innovations, including Structured Latent Space for imposing a desired channel-wise structure on the latent space and Augmented Diffusion Training to utilize the latent space structure to accelerate object structure learning. These techniques collectively enable faster diffusion model convergence and lead to better image generation results, paving the way for employing autoencoders with higher quality upper bound or higher spatial compression ratios.

References

[1] Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, et al. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, pages 929–947, 2024. 5

[2] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22669–22679, 2023. 3, 5, 6, 7

[3] Han Cai, Junyan Li, Muyan Hu, Chuang Gan, and Song Han. Efficientvit: Lightweight multi-scale attention for high-resolution dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17302–17313, 2023.

[4] Han Cai, Muyang Li, Qinsheng Zhang, Ming-Yu Liu, and Song Han. Condition-aware neural network for controlled image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7194–7203, 2024. 3

[5] Hao Chen, Ze Wang, Xiang Li, Ximeng Sun, Fangyi Chen, Jiang Liu, Jindong Wang, Bhiksha Raj, Zicheng Liu, and Emad Barsoum. Softvq-vae: Efficient 1-dimensional continuous tokenizer. *arXiv preprint arXiv:2412.10958*, 2024. 2

[6] Hao Chen, Yujin Han, Fangyi Chen, Xiang Li, Yidong Wang, Jindong Wang, Ze Wang, Zicheng Liu, Difan Zou, and Bhiksha Raj. Masked autoencoders are effective tokenizers for diffusion models. *arXiv preprint arXiv:2502.03444*, 2025.

[7] Junyu Chen, Han Cai, Junsong Chen, Enze Xie, Shang Yang, Haotian Tang, Muyang Li, Yao Lu, and Song Han. Deep compression autoencoder for efficient high-resolution diffusion models. *arXiv preprint arXiv:2410.10733*, 2024. 1, 2, 3, 4, 5, 6, 7, 8

[8] Junsong Chen, YU Jincheng, GE Chongjian, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *International Conference on Learning Representations*, 2024. 3

[9] Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv preprint arXiv:2011.10650*, 2020. 2

[10] Xiaoliang Dai, Ji Hou, Chih-Yao Ma, Sam Tsai, Jialiang Wang, Rui Wang, Peizhao Zhang, Simon Vandenhende, Xiaofang Wang, Abhimanyu Dubey, et al. Emu: Enhancing image generation models using photogenic needles in a haystack. *arXiv preprint arXiv:2309.15807*, 2023. 1, 2

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6, 1

[12] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024. 1, 2, 3

[13] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2, 6

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 5

[15] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24174–24184, 2024. 7

[16] Theodoros Kouzelis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Eq-vae: Equivariance regularized latent space for improved generative image modeling. *arXiv preprint arXiv:2502.09509*, 2025. 2

[17] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019. 6, 1

[18] Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *arXiv preprint arXiv:2404.07724*, 2024. 7

[19] Black Forest Labs. Flux. *Online*, 2024. 1, 2, 7

[20] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *arXiv preprint arXiv:2406.11838*, 2024. 7

[21] Bingchen Liu, Ehsan Akhgari, Alexander Visheratin, Aleks Kamko, Linmiao Xu, Shivam Shrirao, Chase Lambert, Joao Souza, Suhail Doshi, and Daiqing Li. Playground v3: Improving text-to-image alignment with deep-fusion large language models. *arXiv preprint arXiv:2409.10695*, 2024. 1, 2

[22] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. *arXiv preprint arXiv:2401.08740*, 2024. 3, 6, 7

[23] Heusel Martin, Ramsauer Hubert, Unterthiner Thomas, Nessler Bernhard, and Hochreiter Sepp. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30:6626–6637, 2017. 6, 1

[24] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 1, 2, 3, 6, 7, 8

[25] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image

synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 2, 7

[26] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 6, 1

[27] Rui Shu and Stefano Ermon. Bit prioritization in variational autoencoders via progressive coding. In *ICML*, pages 20141–20155, 2022. 2

[28] Ivan Skorokhodov, Sharath Girish, Benran Hu, Willi Menapace, Yanyu Li, Rameen Abdal, Sergey Tulyakov, and Aliaksandr Siarohin. Improving the diffusability of autoencoders. *arXiv preprint arXiv:2502.14831*, 2025. 2

[29] Yuhta Takida, Yukara Ikemiya, Takashi Shibuya, Kazuki Shimada, Woosung Choi, Chieh-Hsin Lai, Naoki Murata, Toshimitsu Uesaka, Kengo Uchida, Wei-Hsiang Liao, et al. Hq-vae: Hierarchical discrete representation learning with variational bayes. *arXiv preprint arXiv:2401.00365*, 2023. 2

[30] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in neural information processing systems*, 33:19667–19679, 2020. 2

[31] Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024. 3

[32] Enze Xie, Junsong Chen, Yuyang Zhao, Jincheng Yu, Ligeng Zhu, Yujun Lin, Zhekai Zhang, Muyang Li, Junyu Chen, Han Cai, et al. Sana 1.5: Efficient scaling of training-time and inference-time compute in linear diffusion transformer. *arXiv preprint arXiv:2501.18427*, 2025. 1

[33] Jingfeng Yao and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. *arXiv preprint arXiv:2501.01423*, 2025. 2

[34] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023. 7

[35] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024. 3

[36] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 5

[37] Zixin Zhu, Xuelu Feng, Dongdong Chen, Jianmin Bao, Le Wang, Yinpeng Chen, Lu Yuan, and Gang Hua. Designing a better asymmetric vqgan for stablediffusion. *arXiv preprint arXiv:2306.04632*, 2023. 2