U-DiTs: Downsample Tokens in U-Shaped Diffusion Transformers

Yuchuan Tian[1*], **Zhijun Tu**[2*], **Hanting Chen**[2], **Jie Hu**[2], **Chao Xu**[1], **Yunhe Wang**[2†]
[1] State Key Lab of General AI, School of Intelligence Science and Technology, Peking University.
[2] Huawei Noah's Ark Lab.
tianyc@stu.pku.edu.cn, {zhijun.tu, chenhanting, hujie23, yunhe.wang}@huawei.com
xuchao@cis.pku.edu.cn

Abstract

Diffusion Transformers (DiTs) introduce the transformer architecture to diffusion tasks for latent-space image generation. With an isotropic architecture that chains a series of transformer blocks, DiTs demonstrate competitive performance and good scalability; but meanwhile, the abandonment of U-Net by DiTs and their following improvements is worth rethinking. To this end, we conduct a simple toy experiment by comparing a U-Net architectured DiT with an isotropic one. It turns out that the U-Net architecture only gain a slight advantage amid the U-Net inductive bias, indicating potential redundancies within the U-Net-style DiT. Inspired by the discovery that U-Net backbone features are low-frequency-dominated, we perform token downsampling on the query-key-value tuple for self-attention that bring further improvements despite a considerable amount of reduction in computation. Based on self-attention with downsampled tokens, we propose a series of U-shaped DiTs (U-DiTs) in the paper and conduct extensive experiments to demonstrate the extraordinary performance of U-DiT models. The proposed U-DiT could outperform DiT-XL/2 with only 1/6 of its computation cost. Codes are available at https://github.com/YuchuanTian/U-DiT.

1 Introduction

Thanks to the attention mechanism that establishes long-range spatial dependencies, Transformers [36] are proved highly effective on various vision tasks including image classification [15], object detection [5], segmentation [43], and image restoration [6]. DiTs [28] introduce full transformer backbones to diffusion, which demonstrate outstanding performance and scalability on image-space and latent-space generation tasks. Recent follow-up works have demonstrated the promising prospect of diffusion transformers by extending their applications to flexible-resolution image generation [26], realistic video generation [2], et cetera.

Interestingly, DiTs have discarded the U-Net architecture [30] that is universally applied in manifold previous works, either in pixel [20; 13] or latent space [29]. The use of isotropic (*i.e.* standard transformer; a plain stack of transformer blocks) architectures in DiTs is indeed successful, as scaled-up DiT models achieve supreme performance. However, the abandonment of the widely-applied U-Net architecture by DiTs and their improvements [18; 10; 26] on latent-space image generation tasks triggers our curiosity, because the U-Net inductive bias is always believed to help denoising. Hence, we rethink deploying DiTs on a canonical U-Net architecture.

In order to experiment with the combination of U-Net with DiT, we first propose a naive DiT in U-Net style (DiT-UNet) and compare it with an isotropic DiT of similar size. Results turn out that

[*]Equal Contribution. [†]Corresponding Author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-1648

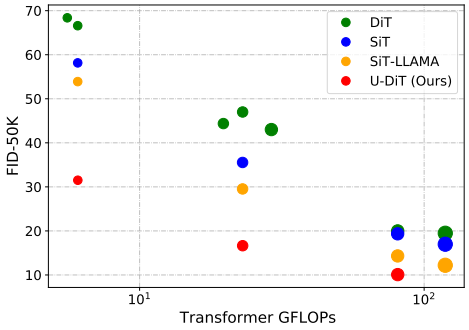


Figure 1: **Comparing U-DiTs with DiTs and their improvements.** We plot FID-50K versus denoiser GFLOPs (in log scale) after 400K training steps. U-DiTs could achieve better performance than its counterparts.

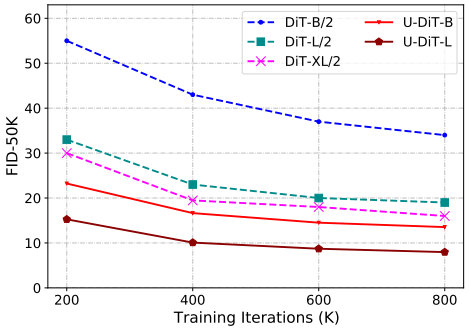


Figure 2: **The performance of U-DiTs and DiTs of various size.** U-DiTs perform consistently better than DiTs with the increase of training steps. The marker size represents the computation cost of the model qualitatively.

DiT-UNets are merely comparable to DiTs at similar computation costs. From this toy experiment, it is inferred that the inductive bias of U-Net is not fully leveraged when U-Nets and plain transformer blocks are simply combined.

Hence, we rethink the self-attention mechanism in DiT-UNet. The backbone in a latent U-Net denoiser provides a feature where low-frequency components dominate [31]. The discovery implies the existence of redundancies in backbone features: the attention module in the U-Net diffuser should highlight low-frequency domains. As previous theories praised downsampling for filtering high-frequency noises in diffusion [39], we seek to leverage this natural low-pass filter by performing token downsampling on the features for self-attention. Unlike previous transformer works [17; 44; 32] that downsample key-value pairs only, we radically downsample the query-key-value tuple altogether, such that self-attention is performed among downsampled latent tokens. It is surprising that when we incorporate self-attention with downsampled tokens into DiT-UNet, better results are achieved on latent U-Net diffusers with a significant reduction of computation.

Based on this discovery, we scale U-Nets with downsampled self-attention up and propose a series of State-of-the-Art U-shaped Diffusion Transformers (**U-DiT**s). We conduct manifold experiments to verify the outstanding performance and scalability of our U-DiT models over isotropic DiTs. As shown in Fig. 1 & Fig. 2, U-DiTs could outperform DiTs by large margins. Amazingly, the proposed U-DiT model could perform better than DiT-XL/2 which is 6 times larger in terms of FLOPs.

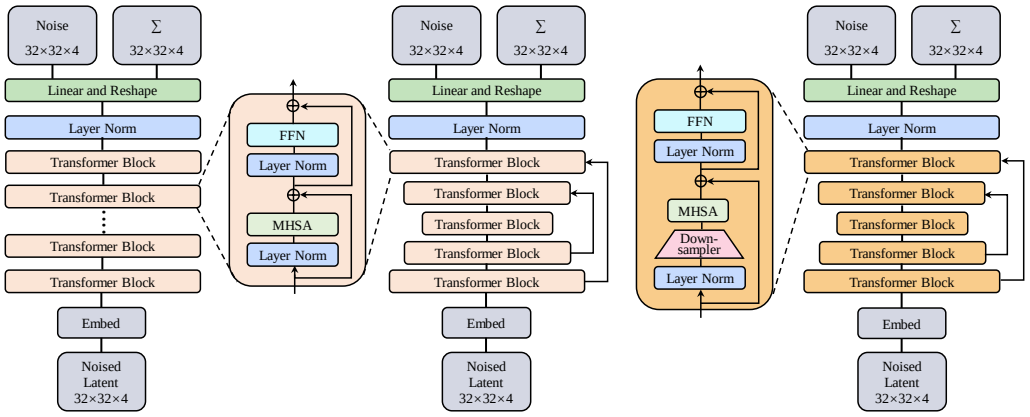
2 Preliminaries

Vision Transformers. ViTs [15] have introduced a transformer backbone to vision tasks by patchifying the input and viewing an image as a sequence of patch tokens and have proved its effectiveness on large-scale image classification tasks. While ViTs adopt an isotropic architecture, some following works on vision transformers [37; 25; 19; 40] adopt a pyramid-like hierarchical architecture that gradually downsamples the feature. The pyramid architecture is proved highly effective in classification and other downstream tasks. Apart from architectural improvements, some other works [3; 41] focuses on improving the Feed-Forward Network module in transformers.

Vision transformers are also mainstream backbones for denoising models. IPT [6] introduces an isotropic transformer backbone for denoising and other low-level tasks. Some later works [23; 22; 9] follow the isotropic convention, but other denoising works [38; 42] shift to U-Net backbones as their design. The pioneering work of U-ViT [1] and DiT [28] introduces full-transformer backbones to diffusion as denoisers.

Recent Advancements in Diffusion Transformers. Following DiTs, some works investigate the training and diffusion [16; 27] strategies of Diffusion Transformers. Other works focus on the design of the DiT backbone. DiffiT [8; 18] introduces a new fusion method for conditions; FiT [26] and VisionLLaMA [10] strengthens DiT by introducing LLM tricks including RoPE2D [34] and

https://doi.org/10.52202/079017-1648



(a) DiT (b) DiT-UNet (c) U-DiT (Ours)

Figure 3: **The evolution from the DiT to the proposed U-DiT.** Left (a): the original DiT, which uses an isotropic architecture. Middle (b): DiT-UNet, which is a plain U-Net-style DiT. We try this as a simple combination of DiT and U-Net in the toy experiment. Right (c): the proposed U-DiT. We propose to downsample the input features for self-attention. The downsampling operation could amazingly improve DiT-UNet with a huge cut on the amount of computation.

SwishGLU. These transformer-based diffusion works agree on adopting isotropic architectures on latents, *i.e.* the latent feature space is not downsampled throughout the whole diffusion model. The authors of DiT [28] even regard the inductive bias of U-Net as "not crucial".

U-Nets for Diffusion. From canonical works [20; 33; 13; 29], the design philosophy of U-Net [30] is generally accepted in diffusion. Specifically, Stable Diffusion [29] uses a U-Net-based denoiser on the compressed latent space for high-resolution image synthesis, which is highly successful in manifold generative tasks. Some previous trials on diffusion transformers [4; 18; 11; 21] also adopt U-Net on pixel-space generation tasks; but strangely, they shifted to isotropic DiT-like structures for latent-space diffusion. Despite its popularity in pixel-space diffusion, the U-Net architecture is not widely accepted in recent transformer-oriented works on latent-space diffusion.

Motivated by this, we are dedicated to investigating the potential of Transformer-backboned U-Net on latent-space diffusion. It is noteworthy that our goal is significantly different from U-ViT [1]: U-ViT is an isotropic transformer architecture with shortcuts, but our work resort to true U-Net architectures that involves multiple stages of feature-map downsampling and upsampling.

3 Investigating U-Net DiTs in Latent

As is recapped, the U-Net architecture is widely adopted in diffusion applications; theoretical evaluations on U-Net denoisers also reveal their advantage, as downsampling U-Net stage transitions could filter noises that dominate high frequencies [39]. The unprecedented desertion of isotropic architectures for latent diffusion transformers is thus counter-intuitive. We are rethinking and elucidating the potentials of transformer-backboned U-Net denoisers in latent diffusion via a toy experiment.

A canonical U-Net-style DiT. To start with, we propose a naive Transformer-backboned U-Net denoiser named **DiT-UNet** by embedding DiT blocks into a canonical U-Net architecture. Following previous U-Net designs, The DiT-UNet consists of an encoder and a decoder with an equal number of stages. When the encoder processes the input image by downsampling the image as stage-level amounts, the decoder scales up the encoded image from the most compressed stage to input size. At each encoder stage transition, spatial downsampling by the factor of 2 is performed while the feature dimension is doubled as well. Skip connections are provided at each stage transition. The skipped feature is concatenated and fused with the upsampled output from the previous decoder stage, replenishing information loss to decoders brought by feature downsampling. Considering the

https://doi.org/10.52202/079017-1648

small, cramped latent space (32× 32 for 256×256-sized generation), we designate 3 stages in total, *i.e.* the feature is downsampled two times and subsequently recovered to its original size. In order to fit time and condition embeddings for various feature dimensions across multiscale stages, we use independent embedders for respective stages. In addition, we avoid patchifying the latent, as the U-Net architecture itself downsamples the latent space and there is no need for further spatial compression.

Via toy experiments, we compare the proposed U-Net-style DiT with the original DiT that adopts an isotropic architecture. In order to align the model with the DiT design, we repeatedly use plain DiT blocks in each stage. Each DiT block includes a self-attention module as the token mixer and a two-layer feed-forward network as the channel mixer. We conduct the experiment by training the U-Net-Style DiT for 400K iterations and compare it with DiT-S/4 which is comparable in size. All training hyperparameters are kept unchanged. It occurs that the U-Net style DiT only gains a limited advantage over the original isotropic DiT. The inductive bias of U-Net is insufficiently utilized.

ImageNet 256×256						
Model	GFLOPs	FID↓	sFID↓	IS↑	Precision↑	Recall↑
DiT-S/4	1.41	97.85	21.19	13.27	0.26	0.41
DiT-UNet	1.40	93.48	**20.41**	14.20	0.27	0.42
DiT-UNet+Key-Value Downsampling	0.91	94.38	23.21	14.32	0.27	0.40
DiT-UNet+**Token Downsampling (Ours)**	**0.90**	**89.43**	21.36	**15.13**	**0.29**	**0.44**

Table 1: **Toy experiments on U-Net-style DiTs.** The naive DiT-UNet performs slightly better than the isotropic DiT-S/4; but interestingly, when we apply token downsampling for self-attention, the DiT-UNet performs better with fewer costs.

Improved U-Net-style DiT via token downsampling. In seeking to incorporate attention in transformers to diffusion U-Nets better, we review the role of the U-Net backbone as the diffusion denoiser. A recent work on latent diffusion models [31] conducted frequency analysis on intermediate features from the U-Net backbone, and concluded that energy concentrates at the low-frequency domain. This frequency-domain discovery hints at potential redundancies in the backbone: the U-Net backbone should highlight the coarse object from a global perspective rather than the high-frequency details.

Naturally, we resort to attention with downsampled tokens. The operation of downsampling is a natural low-pass filter that discards high-frequency components. The low-pass feature of downsampling has been investigated under the diffusion scenario, which concludes that downsampling helps denoisers in diffusion as it automatically "discards those higher-frequency subspaces which are dominated by noise" [39]. Hence, we opt to downsample tokens for attention.

In fact, attention to downsampled tokens is not new. Previous works regarding vision transformers [17; 44] have proposed methods to downsample key-value pairs for computation cost reduction. Recent work on acceleration of diffusion models [32; 7] also applies key-value downsampling on Stable Diffusion models. But these works maintain the number of queries, and thus the downsampling operation is not completely performed. Besides, these downsampling measures usually involves a reduction of tensor size, which could result in a significant loss in information.

Different from these works, we propose a simple yet radical token downsampling method for DiT-UNets: we downsample queries, keys, and values at the same time for diffusion-friendly self-attention, but meanwhile we keep the overall tensor size to avoid information loss. The procedure is detailed as follows: the feature-map input is first converted into four $2\times$ downsampled features by the downsampler (the downsampler design is detailed in Sec. 4.2). Then, the downsampled features are mapped to Q, K, V for self-attention. Self-attention is performed within each downsampled feature. After the attention operation, the downsampled tokens are spatially merged as a unity to recover the original number of tokens. Notably, the feature dimension is kept intact during the whole process. Unlike U-Net downsampling, we are not reducing or increasing the number of elements in the feature during the downsampling process. Rather, we send four downsampled tokens into self-attention in a parallel manner.

Self-attention with downsampled tokens does help DiT-UNets on the task of latent diffusion. As shown in Tab. 1, the substitution of downsampled self-attention to full-scale self-attention brings

slight improvement in the Fréchet Inception Distance (FID) metric despite a significant reduction in FLOPs.

Complexity analysis. Apart from the performance benefits, we are aware that adopting downsampled self-attention in the U-Shaped DiT could save as much as 1/3 of the model's overall computation cost. We conduct a brief computation complexity analysis on the self-attention mechanism to explain where the savings come from.

Given an input feature of size $N \times N$ and dimension d, we denote $Q, K, V \in \mathbb{R}^{N^2 \times d}$ as mapped query-key-value tuples. The complexity of self-attention is analyzed as:

$$X = \underbrace{AV}_{\mathcal{O}(N^4 D)} \qquad \text{s.t.} \qquad A = \textbf{Softmax}\underbrace{\left(QK^T\right)}_{\mathcal{O}(N^4 D)}.$$

In the proposed self-attention on downsampled tokens, four sets of downsampled query-key-value tuples $4 \times (Q_{\downarrow 2}, K_{\downarrow 2}, V_{\downarrow 2}) \in \mathbb{R}^{(\frac{N}{2})^2 \times d}$ performs self-attention respectively. While each self-attention operation costs only 1/16 of full-scale self-attention, the total cost for downsampled self-attention is 1/4 of full-scale self-attention. 3/4 of the self-attention computation is saved via token downsampling.

In a nutshell, we show from toy experiments that the redundancy of DiT-UNet is reduced by downsampling the tokens for self-attention.

4 Scaling the Model Up

Based on the discovery in our toy experiment, we propose a series of U-shaped DiTs (**U-DiT**) by applying the downsampled self-attention (proposed in Sec. 3) and scaling U-Net-Style DiT up.

Settings. We adopt the training setting of DiT. The same VAE (*i.e.* sd-vae-ft-ema) for latent diffusion models [29] and the AdamW optimizer is adopted. The training hyperparameters are kept unchanged, including global batch size 256, learning rate $1e-4$, weight decay 0, and global seed 0. The training is conducted with the training set of ImageNet 2012 [12]. We used 8 NVIDIA A100s (80G) to train U-DiT-B and U-DiT-L models. The training overhead is listed in the appendix.

Apart from the self-attention on downsampling as introduced in the toy experiment (Section 3), we further introduce a series of modifications to U-DiTs, including cosine similarity attention [24; 22], RoPE2D [34; 26; 10], depthwise conv FFN [38; 3; 44], and re-parametrization [14; 35]. The contribution of each modification is quantitatively evaluated in Sec. 9.

4.1 U-DiT at Larger Scales

Comparison with DiTs and their improvements. In order to validate the effectiveness of the proposed U-DiT models beyond simple toy experiments, we scale them up and compare them with DiTs [28] of larger sizes. For a fair comparison, we use the same sets of training hyperparameters as DiT; all models are trained for 400K iterations. The results on ImageNet 256×256 are shown in Tab. 2, where we scale U-DiTs to $\sim 6e9$, $\sim 20e9$, $\sim 80e9$ FLOPs respectively and compare them with DiTs of similar computation costs, more details about the U-DiT architectures are shown in Tab. 8.

It could be concluded from Tab. 2 that all U-DiT models could outcompete their isotropic counterparts by considerable margins. Specifically, U-DiT-S and U-DiT-B could outperform DiTs of comparable size by ~ 30 FIDs; U-DiT-L could outperform DiT-XL/2 by ~ 10 FIDs. It is shocking that U-DiT-B could outcompete DiT-XL/2 with only 1/6 of the computation costs. In Tab. 3, we further demonstrate the advantage of U-DiTs over several competitive diffusion transformers [1; 28; 8; 18]. To present the advantage of our method better, we also include the performance of U-DiTs in an FID-50K versus FLOPs plot (Fig. 1). Apart from DiTs and U-DiTs, we also include other state-of-the-art methods: SiT [27] that proposes an interpolant framework for DiTs, and SiT-LLaMA [10] that combines state-of-the-art DiT backbone VisionLLaMA and SiT. The advantages of U-DiTs over other baselines are prominent in the plot. The results highlight the extraordinary scalability of the proposed U-DiT models.

https://doi.org/10.52202/079017-1648

ImageNet 256×256						
Model	FLOPs(G)	FID↓	sFID↓	IS↑	Precision↑	Recall↑
DiT-S/2 [28]	6.06	68.40	-	-	-	-
DiT-S/2*	6.07	67.40	11.93	20.44	0.368	0.559
U-DiT-S (Ours)	6.04	**31.51**	**8.97**	**51.62**	**0.543**	**0.633**
DiT-L/4 [28]	19.70	45.64	-	-	-	-
DiT-L/4*	19.70	46.10	9.17	31.05	0.472	0.612
DiT-B/2 [28]	23.01	43.47	-	-	-	-
DiT-B/2*	23.02	42.84	8.24	33.66	0.491	0.629
U-DiT-B (Ours)	22.22	**16.64**	**6.33**	**85.15**	**0.642**	**0.639**
DiT-L/2 [28]	80.71	23.33	-	-	-	-
DiT-L/2*	80.75	23.27	6.35	59.63	0.611	**0.635**
DiT-XL/2 [28]	118.64	19.47	-	-	-	-
DiT-XL/2*	118.68	20.05	6.25	66.74	0.632	0.629
U-DiT-L (Ours)	85.00	**10.08**	**5.21**	**112.44**	**0.702**	0.631

Table 2: **Comparing U-DiTs against DiTs on ImageNet 256×256 generation.** Experiments with supermarks * are replicated according to the official code of DiT. We compare models trained for 400K iterations with the standard training hyperparameters of DiT. The performance of U-DiTs is outstanding: U-DiT-B could beat DiT-XL/2 with only **1/6** of inference FLOPs; U-DiT-L could outcompete DiT-XL/2 by 10 FIDs.

ImageNet 256×256						
Model	FLOPs (G)	FID↓	sFID↓	IS↑	Precision↑	Recall↑
U-ViT-L [1]	76.4	21.22	6.10	67.64	0.615	0.633
U-ViT-XL* [1]	113.0	18.35	5.75	76.59	0.632	0.630
DiT-XL/2 [28]	118.7	20.05	6.25	66.74	0.632	0.629
PixArt-α-XL/2* [8]	118.4	24.75	6.08	52.24	0.612	0.613
DiffiT-XL/2* [18]	118.5	36.86	6.53	35.39	0.540	0.613
U-DiT-B (Ours)	22.2	16.64	6.33	85.15	0.642	**0.639**
U-DiT-L (Ours)	85.0	**10.08**	**5.21**	**112.44**	**0.702**	0.631

Table 3: **Comparing U-DiTs against competitive diffusion architectures on ImageNet 256×256 generation.** Since different architectures use different training settings, we align them under the official 400K-iteration setting of DiT for a fair comparison. The proposed U-DiT series could outperform these models by large margins at fewer FLOPs. Experiments with supermarks * include necessary modifications of the original work (detailed in the appendix).

U-DiTs are also performant in generation scenarios with classifier-free guidance. In Tab. 4, we compare U-DiTs with DiTs at $cfg = 1.5$. For a fair comparison, we train U-DiTs and DiTs for 400K iterations under identical settings.

Extended training steps. We evacuate the potentials of U-DiTs by extending training steps to 1 Million. Fig. 2 further demonstrate that the advantage of U-DiTs is consistent at all training steps. As training steps gradually goes up to 1 Million, the performance of U-DiTs is improving (Tab. 5). We

ImageNet 256×256							
Model	Cfg-Scale	FLOPs(G)	FID↓	sFID↓	IS↑	Precision↑	Recall↑
DiT-L/2*	1.5	80.75	7.53	4.78	134.69	0.780	**0.532**
DiT-XL/2*	1.5	118.68	6.24	4.66	150.10	0.794	0.514
U-DiT-B	1.5	22.22	4.26	4.74	199.18	0.825	0.507
U-DiT-L	1.5	85.00	**3.37**	**4.49**	**246.03**	**0.862**	0.502

Table 4: **Generation performance with classifier-free guidance.** We measure the performance of U-DiTs and DiTs at 400K training steps with $cfg = 1.5$. Experiments with a supermark * are replicated according to the official code of DiT. U-DiTs are also performant on conditional generation.

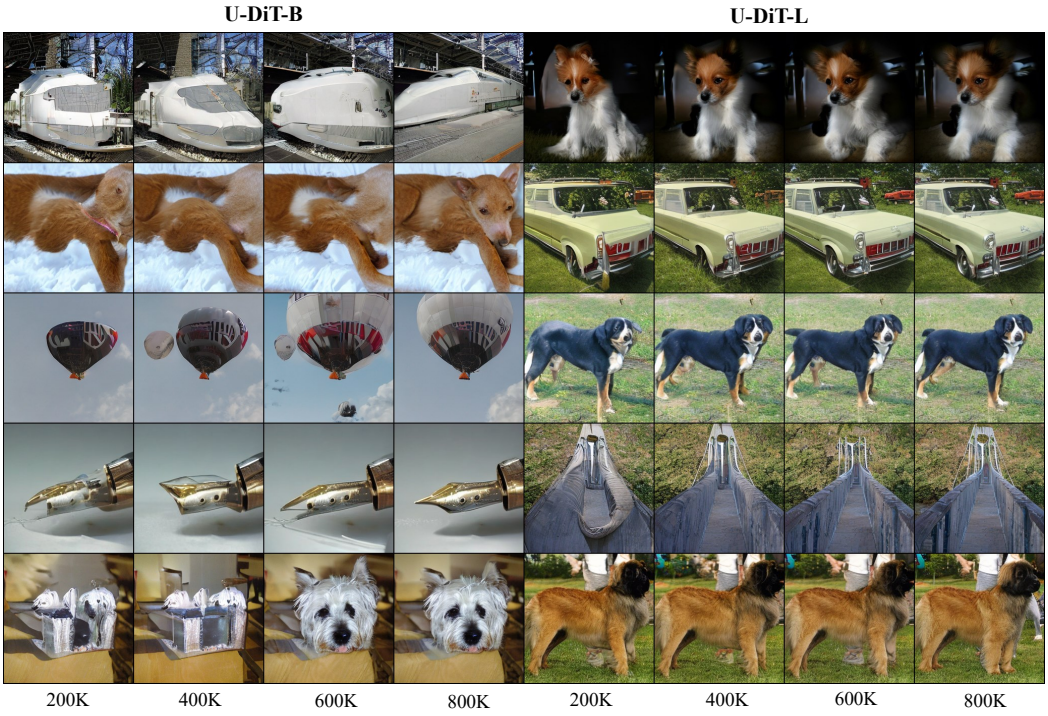


Figure 4: **Quality improvements of generated samples as training continues.** We sample from U-DiT models trained for different numbers of iterations on ImageNet 256×256. More training does improve generation quality. Best viewed on screen.

visualize the process where the image quality is gradually getting better (Fig. 4). Notably, U-DiT-L at only 600K training steps could outperform DiT-XL/2 at 7M training steps without classifier-free guidance. As additionally shown in Fig. 5, U-DiT models could conditionally generate authentic images at merely 1M iterations.

Larger image size. We additionally compare the generation performance of U-DiT-B and DiT-XL/2 on ImageNet 512 × 512 under exactly the same training setting. As shown in Tab. 6, U-DiT-B could still outcompete DiT-XL/2 that is approximately 5 times larger in FLOPs.

ImageNet 256×256						
Model	Training Steps	FID↓	sFID↓	IS↑	Precision↑	Recall↑
DiT-XL/2	7M	9.62	-	-	-	-
U-DiT-B	200K	23.23	6.84	64.42	0.610	0.621
U-DiT-B	400K	16.64	6.33	85.15	0.642	0.639
U-DiT-B	600K	14.51	6.30	94.56	0.652	0.643
U-DiT-B	800K	13.53	**6.27**	98.99	0.654	0.645
U-DiT-B	1M	**12.87**	6.33	**103.79**	**0.661**	**0.653**
U-DiT-L	200K	15.26	5.60	86.01	0.685	0.615
U-DiT-L	400K	10.08	5.21	112.44	0.702	0.631
U-DiT-L	600K	8.71	**5.17**	122.45	0.705	0.645
U-DiT-L	800K	7.96	5.21	131.35	0.705	0.648
U-DiT-L	1M	**7.54**	5.27	**135.49**	**0.706**	**0.659**

Table 5: **The performance of U-DiT-B and U-DiT-L models with respect to training iterations.** The unconditional generation performance of both models on ImageNet 256×256 consistently improves as training goes on, where U-DiT-L at 600K steps strikingly beats DiT-XL/2 at 7M steps.

https://doi.org/10.52202/079017-1648

ImageNet 512×512						
Model	FLOPs (G)	FID↓	sFID↓	IS↑	Precision↑	Recall↑
DiT-XL/2*	524.7	20.94	**6.78**	66.30	0.745	0.581
U-DiT-B	106.7	**15.39**	6.86	**92.73**	**0.756**	**0.605**

Table 6: **Comparing U-DiTs against DiTs on ImageNet 512×512 generation.** Experiments with a supermark * are replicated according to the official code of DiT. We compare models trained for 400K iterations with the standard training hyperparameters of DiT.

ImageNet 256×256						
Model	FLOPs(G)	FID↓	sFID↓	IS↑	Precision↑	Recall↑
Pixel Shuffle (PS)	0.89	96.15	23.90	13.93	0.272	0.389
Depthwise (DW) Conv. + PS	0.91	89.87	**20.99**	14.92	0.288	0.419
DW Conv. ‖ Shortcut + PS	0.91	**89.43**	21.36	**15.13**	**0.291**	**0.436**

Table 7: **Ablations on the choice of downsampler.** We have tried several downsampler designs, and it turns out that the parallel connection of a shortcut and a depthwise convolution is the best fit. We avoid using ordinary convolution (*i.e.* Conv.+PS) because channel-mixing is costly: conventional convolution-based downsamplers could double the amount of computation. The U-DiT with a conventional downsampler costs as many as 2.22G FLOPs in total.

4.2 Ablations

The design of downsampler. The downsampling operation in the proposed U-DiT transforms a complete feature into multiple spatially downsampled features. Based on previous wisdom, we figured out that previous works either directly perform pixel shuffling, or apply a convolution layer before pixel shuffling. While we hold that it is much too rigid to shuffle pixels directly as downsampling, applying convolution is hardly affordable in terms of computation costs. Specifically, ordinary convolutions are costly as extensive dense connections on the channel dimension are involved: using convolution-based downsamplers could double computation costs. As a compromise, we apply depthwise convolution instead. We also add a shortcut that short-circuits this depthwise convolution, which has proved crucial for better performance. The shortcut adds negligible computation cost to the model, and in fact, it could be removed during the inference stage with re-parameterization tricks. The results are shown in Tab. 7.

The contribution of each individual modification. In this part, we start from a plain U-Net-style DiT (DiT-UNet) and evaluate the contribution of individual components. Firstly, we inspect the advantage of downsampled self-attention. Recapping the toy experiment results in Sec. 3, replacing the full-scale self-attention with downsampled self-attention would result in an improvement in FID and 1/3 reduction in FLOPs. In order to evaluate the improvement of downsampling via model performance, we also design a slim version of DiT-UNet (*i.e.* DIT-UNet (Slim)). The DiT-UNet (Slim) serves as a full-scale self-attention baseline that spends approximately the same amount (~ 0.9GFLOPs) of computation as our U-DiT. As shown in the upper part of Tab. 9, by comparing U-DiT against DiT-UNet (Slim), it turns out that downsampling tokens in DiT-UNet could bring a performance improvement of ~ 18FIDs.

Next, we inspect other modifications that further refine U-DiTs (lower part of Tab. 9). Swin Transformer V2 [24] proposes a stronger variant of self-attention: instead of directly multiplying Q and K matrices, cosine similarities between queries and keys are used. We apply the design to our self-attention, which yields ~ 2.5FIDs of improvement. RoPE [34] is a powerful positional embedding method, which has been widely applied in Large Language Models. Following the latest diffusion transformer works [26; 10], we inject 2-dimensional RoPE (RoPE2D) into queries and keys right before self-attention. The introduction of RoPE2D improves performance by ~ 2.5FIDs. Some recent transformer works strengthen MLP by inserting a depthwise convolution layer between two linear mappings [38; 3; 44]. As the measure is proved effective in these works, we borrow it to our U-DiT model, improving ~ 5FIDs. As re-parametrization during training [14] could improve model performance, we apply the trick to FFN [35] and bring an additional improvement of ~ 3.5FIDs. Above all, based on these components, the proposed U-DiTs are further improved.

Apart from the modifications that improve U-DiT, it is worth noting that vanilla U-DiTs (*i.e.* U-DiTs without any of the modifications mentioned above) are still competitive. According to Tab. 10, vanilla U-DiT-L could still achieve $\sim$ 8FIDs of advantage over DiT-XL/2.

Model	Params (M)	FLOPs (G)	Channel	Head Number	Encoder-Decoder
U-DiT-S	52.05	6.04	96	4	[2, 5, 8, 5, 2]
U-DiT-B	204.43	22.22	192	8	[2, 5, 8, 5, 2]
U-DiT-L	810.19	85.00	384	16	[2, 5, 8, 5, 2]

Table 8: **Configurations of U-DiTs architecture with different model sizes.** Channel represents the initial output channel number of first layer. Encoder-Decoder denotes the transformer block number of encoder and decoder module.

ImageNet 256×256						
Model	FLOPs(G)	FID↓	sFID↓	IS↑	Precision↑	Recall↑
DiT-UNet (Slim)	0.92	107.00	24.66	11.95	0.230	0.315
DiT-UNet	1.40	93.48	20.41	14.20	0.274	0.415
U-DiT-T (DiT-UNet+Downsampling)	**0.91**	89.43	21.36	15.13	0.291	0.436
U-DiT-T (+Cos.Sim.)	0.91	86.96	19.98	15.63	0.299	0.450
U-DiT-T (+RoPE2D)	0.91	84.64	19.38	16.19	0.306	0.454
U-DiT-T (+DWconv FFN)	0.95	79.30	17.84	17.48	0.326	0.494
U-DiT-T (+Re-param.)	0.95	**75.71**	**16.27**	**18.59**	**0.336**	**0.512**

Table 9: **Ablations on U-DiT components.** Apart from the toy example in Sec. 3, we further validate the effectiveness of downsampled by comparing the U-DiT with a slimmed version of DiT-UNet at equal FLOPs. Results reveal that downsampling could bring $\sim$ 18FIDs on DiT-UNet. Further modifications on top of the U-DiT architecture could improve 2 to 5 FIDs each.

ImageNet 256×256						
Model	FLOPs(G)	FID↓	sFID↓	IS↑	Precision↑	Recall↑
U-DiT-S (Vanilla)	5.91	41.01	10.96	39.29	0.489	0.622
U-DiT-S (+All Mods)	6.04	**31.51**	**8.97**	**51.62**	**0.543**	**0.633**
U-DiT-B (Vanilla)	21.96	20.89	7.33	72.85	0.611	0.637
U-DiT-B (+All Mods)	22.22	**16.64**	**6.33**	**85.15**	**0.642**	**0.639**
U-DiT-L (Vanilla)	84.48	12.04	5.37	102.63	0.684	0.628
U-DiT-L (+All Mods)	85.00	**10.08**	**5.21**	**112.44**	**0.702**	**0.631**

Table 10: **Comparison between vanilla U-DiTs and improved U-DiTs with all modifications.** With negligible extra computational overhead, the proposed modifications could improve the performance of U-DiT; but it is worth noting that vanilla U-DiTs are powerful enough against DiTs.

5 Conclusion

In this paper, we lay emphasis on DiTs in U-Net architecture for latent-space generation. Though isotropic-architectured DiTs have proved their strong scalability and outstanding performance, the effectiveness of the U-Net inductive bias is neglected. Thus, we rethink DiTs in the U-Net style. We first conduct an investigation on plain DiT-UNet, which is a straightforward combination of U-Net and DiT blocks, and try to reduce computation redundancy in the U-Net backbone. Inspired by previous wisdom on diffusion, we propose to downsample the visual tokens for self-attention and yield extraordinary results: the performance is further improved despite a huge cut on FLOPs. From this interesting discovery, we scale the U-Net architecture up and propose a series of U-shaped DiT models (U-DiTs). We have done various experiments to demonstrate the outstanding performance and scalability of our U-DiTs.

Limitations. For lack of computation resources and tight schedule, at this time we could not further extend training iterations and scale the model size up to fully investigate the potential of U-DiTs.

https://doi.org/10.52202/079017-1648

Figure 5: **Generated samples by U-DiT-L at 1M iterations.** It is astonishing that U-DiT could achieve authentic visual quality at merely 1 Million training steps. Best viewed on screen.

Broader Impacts. Due to the biases in the training data set, the generated content may contain pornographic, racist, hate and violent information. But we emphasize that the potential for misuse is mitigated through vigilant application.

Discussion of Safeguards. For cautionous usage, we suggest an algorithm capable of checking generated images, in order to identify and mitigate content that contravenes legal or ethical usages.

Acknowledgement. This work is supported by the National Key R&D Program of China under grant No. 2022ZD0160300 and the National Natural Science Foundation of China under grant No. 62276007. We gratefully acknowledge the support of MindSpore, CANN, and Ascend AI Processor used for this research.

References

[1] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 22669–22679. IEEE, 2023.

[2] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024.

[3] Han Cai, Junyan Li, Muyan Hu, Chuang Gan, and Song Han. Efficientvit: Multi-scale linear attention for high-resolution dense prediction. *arXiv preprint arXiv:2205.14756*, 2022.

[4] He Cao, Jianan Wang, Tianhe Ren, Xianbiao Qi, Yihao Chen, Yuan Yao, and Lei Zhang. Exploring vision transformers as diffusion learners. *CoRR*, abs/2212.13771, 2022.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, volume 12346 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2020.

[6] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. *CoRR*, abs/2012.00364, 2020.

[7] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-Σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. *CoRR*, abs/2403.04692, 2024.

[8] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis, 2023.

[9] Xiangyu Chen, Xintao Wang, Wenlong Zhang, Xiangtao Kong, Yu Qiao, Jiantao Zhou, and Chao Dong. Hat: Hybrid attention transformer for image restoration. *arXiv preprint arXiv:2309.05239*, 2023.

[10] Xiangxiang Chu, Jianlin Su, Bo Zhang, and Chunhua Shen. Visionllama: A unified llama interface for vision tasks. *CoRR*, abs/2403.00522, 2024.

[11] Katherine Crowson, Stefan Andreas Baumann, Alex Birch, Tanishq Mathew Abraham, Daniel Z. Kaplan, and Enrico Shippole. Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. *CoRR*, abs/2401.11605, 2024.

[12] 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 Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society, 2009.

[13] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat gans on image synthesis. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 8780–8794, 2021.

[14] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13733–13742, 2021.

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020.

[16] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Mdtv2: Masked diffusion transformer is a strong image synthesizer, 2024.

[17] Jianyuan Guo, Kai Han, Han Wu, Chang Xu, Yehui Tang, Chunjing Xu, and Yunhe Wang. CMT: convolutional neural networks meet vision transformers. *CoRR*, abs/2107.06263, 2021.

[18] Ali Hatamizadeh, Jiaming Song, Guilin Liu, Jan Kautz, and Arash Vahdat. Diffit: Diffusion vision transformers for image generation. *CoRR*, abs/2312.02139, 2023.

[19] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 11916–11925. IEEE, 2021.

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239, 2020.

[21] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 13213–13232. PMLR, 2023.

[22] Yawei Li, Yuchen Fan, Xiaoyu Xiang, Denis Demandolx, Rakesh Ranjan, Radu Timofte, and Luc Van Gool. Efficient and explicit modelling of image hierarchies for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18278–18289, 2023.

[23] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 1833–1844, October 2021.

 https://doi.org/10.52202/079017-1648

[24] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12009–12019, 2022.

[25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, October 2021.

[26] Zeyu Lu, Zidong Wang, Di Huang, Chengyue Wu, Xihui Liu, Wanli Ouyang, and Lei Bai. Fit: Flexible vision transformer for diffusion model. *CoRR*, abs/2402.12376, 2024.

[27] 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. *CoRR*, abs/2401.08740, 2024.

[28] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 4172–4182. IEEE, 2023.

[29] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10674–10685. IEEE, 2022.

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells III, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer, 2015.

[31] Chenyang Si, Ziqi Huang, Yuming Jiang, and Ziwei Liu. Freeu: Free lunch in diffusion u-net. *CoRR*, abs/2309.11497, 2023.

[32] Ethan Smith, Nayan Saxena, and Aninda Saha. Todo: Token downsampling for efficient generation of high-resolution images, 2024.

[33] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *CoRR*, abs/2010.02502, 2020.

[34] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[35] Zhijun Tu, Kunpeng Du, Hanting Chen, Hailing Wang, Wei Li, Jie Hu, and Yunhe Wang. Ipt-v2: Efficient image processing transformer using hierarchical attentions, 2024.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[37] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *CoRR*, abs/2102.12122, 2021.

[38] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17683–17693, 2022.

[39] Christopher Williams, Fabian Falck, George Deligiannidis, Chris C. Holmes, Arnaud Doucet, and Saifuddin Syed. A unified framework for u-net design and analysis. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[40] Yixing Xu, Chao Li, Dong Li, Xiao Sheng, Fan Jiang, Lu Tian, and Ashish Sirasao. Fdvit: Improve the hierarchical architecture of vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5950–5960, October 2023.

[41] Yixing Xu, Chao Li, Dong Li, Xiao Sheng, Fan Jiang, Lu Tian, Ashish Sirasao, and Emad Barsoum. Enhancing vision transformer: Amplifying non-linearity in feedforward network module. In *Forty-first International Conference on Machine Learning*, 2024.

[42] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, 2022.

[43] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H. S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *CoRR*, abs/2012.15840, 2020.

[44] Yupeng Zhou, Zhen Li, Chun-Le Guo, Song Bai, Ming-Ming Cheng, and Qibin Hou. Srformer: Permuted self-attention for single image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12780–12791, 2023.

 https://doi.org/10.52202/079017-1648

A Appendix / supplemental material

A.1 Details about Downsampling

Given an input tuple of (queries, keys, values) QKV (shape=$(b, 3c, h, w)$), we firstly conduct Pixel-UnShuffle operation on QKV, and get four spatially downsampled QKV (shape=$4 \times (bs^2, 3c, h/s, w/s)$). Then we perform vanilla multi-head self-attention, and get four downsampled output (shape=$4 \times (b \times s^2, c, h/s, w/s)$). Finally, we merge the four downsampled outputs into unity via Pixel-Shuffling (shape=$(b, c, h/s, w/s)$). Throughout the process, we not only significantly reduced the computational overhead of self-attention, but also ensured that the entire upsampling and downsampling process was completely lossless: the feature maps have not gone through lossy downsampling like bicubic or bilinear downsampling.

A.2 Additional Experiment Details

Training Overhead. We report the training speed in Table 11. The training speed of vanilla U-DiT-L is comparable to that of DiT-XL/2.

ImageNet 256×256						
Model	TS (Steps/Sec)	FID↓	sFID↓	IS↑	Precision↑	Recall↑
DiT-XL/2* [28]	1.71	20.05	6.25	66.74	0.632	0.629
U-DiT-B (Vanilla)	3.14	20.89	7.33	72.85	0.611	0.637
U-DiT-L (Vanilla)	1.55	12.04	5.37	102.63	0.684	0.628
U-DiT-L (+All Mods)	0.84	**10.08**	**5.21**	**112.44**	**0.702**	**0.631**

Table 11: **The training overhead of DiT-XL/2 and U-DiTs.** "TS" stands for training speed, measured in steps per second on 8 NVIDIA A100 (80G).

Experiment Details in Table 3. Since different diffusion architectures use different settings, we are dedicated to comparing them under identical settings for fair comparison. We adopt the 400K-iteration training setting of DiT-XL/2 [28]. Here are some further details regarding certain baselines:

1. **U-ViT-XL**: We increase the depth of U-ViT-L from 20 to 30 in order to match the FLOPs of DiT-XL/2. We encounter loss explosion while training U-ViT-H (133.25 GFLOPs) on the codebase of DiTs.
2. **PixArt-α-XL/2**: As the original model is a text-to-image model, we removed its cross attention module for texts.
3. **DiffiT-XL/2**: This model is not open-sourced at the moment of this publication. Since it is a variant of DiT-XL/2, we replicated the time-dependent self-attention (TMSA) based on the codes of DiT. Unfortunately, the performance gets worse compared to the original DiT-XL/2.

Additional Visual Results. Due to large file size, we are unable to provide all visual results in the appendix. Please refer to the supplementary materials for two high-quality visual result demos.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and precede the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract demonstrates our motivation, the proposed ideas and a brief summary of experiment results.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper has discussed the limitations of the work.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.

https://doi.org/10.52202/079017-1648

- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The paper does not inlcude theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper fully discloses all the information needed to reproduce the main experimental results of the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper will provide open access to the data and code during camera ready period.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: This paper has specified all the training and test details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: This is not relevant to this paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

https://doi.org/10.52202/079017-1648

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper has indicated sufficient information on the computer resources.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: This paper has discussed both potential positive societal impacts and negative societal impacts of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: This paper has described safeguards.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The utilization of code, data and models in this paper is in accordance with the license and the terms.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

https://doi.org/10.52202/079017-1648

Justification: This paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.