ICML
International Conference
On Machine Learning

FiT: Flexible Vision Transformer for Diffusion Model

Zeyu Lu[1,2], Zidong Wang[2,3], Di Huang[2,4], Chengyue Wu[5],
Xihui Liu[5], Wanli Ouyang[2], Lei Bai[2]

[1]Shanghai Jiao Tong University
[2]Shanghai Artificial Intelligence Laboratory [3]Tsinghua University
[4]The University of Sydney [5]The University of Hong Kong



Overview

FiT is an novel transformer architecture which is capable of generating images at unrestricted resolutions and aspect ratios.

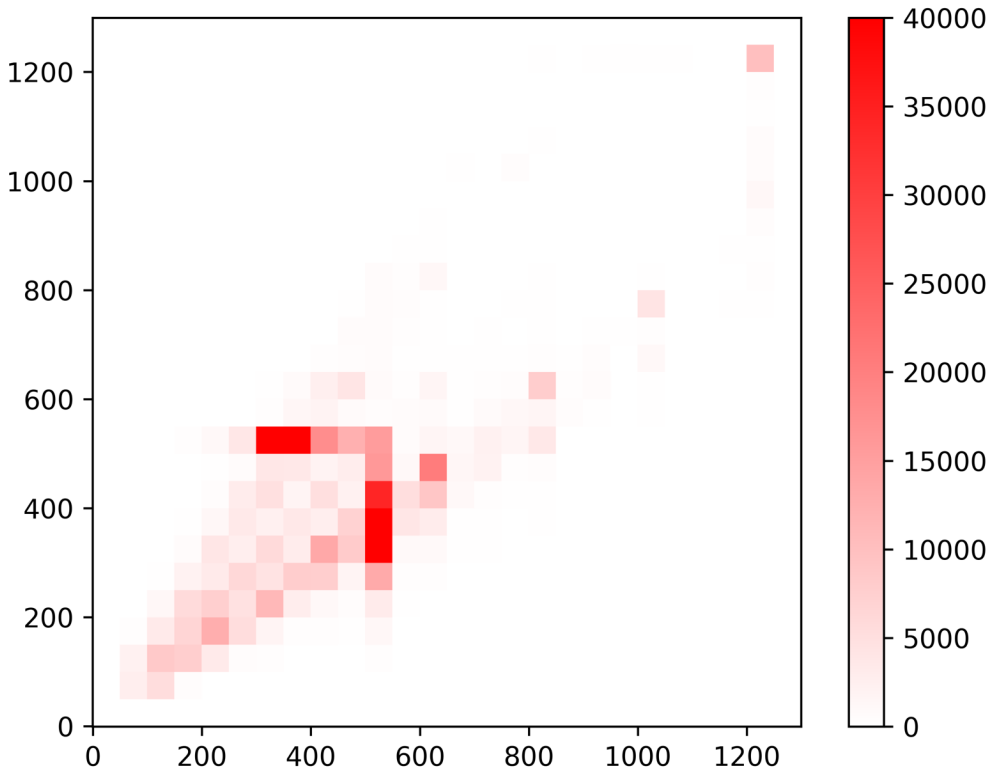
Selected samples from FiT-XL/2 models at resolutions of 256 × 256, 224 × 448 and 448 × 224.

Motivation

Nowadays, training a generative model often requires a large number of images.

Most of these images come from web crawled images, so they have **complex resolutions and aspect ratios**.

We have shown the Height/Width distribution of *ImageNet1K* as shown in the figure on the right.



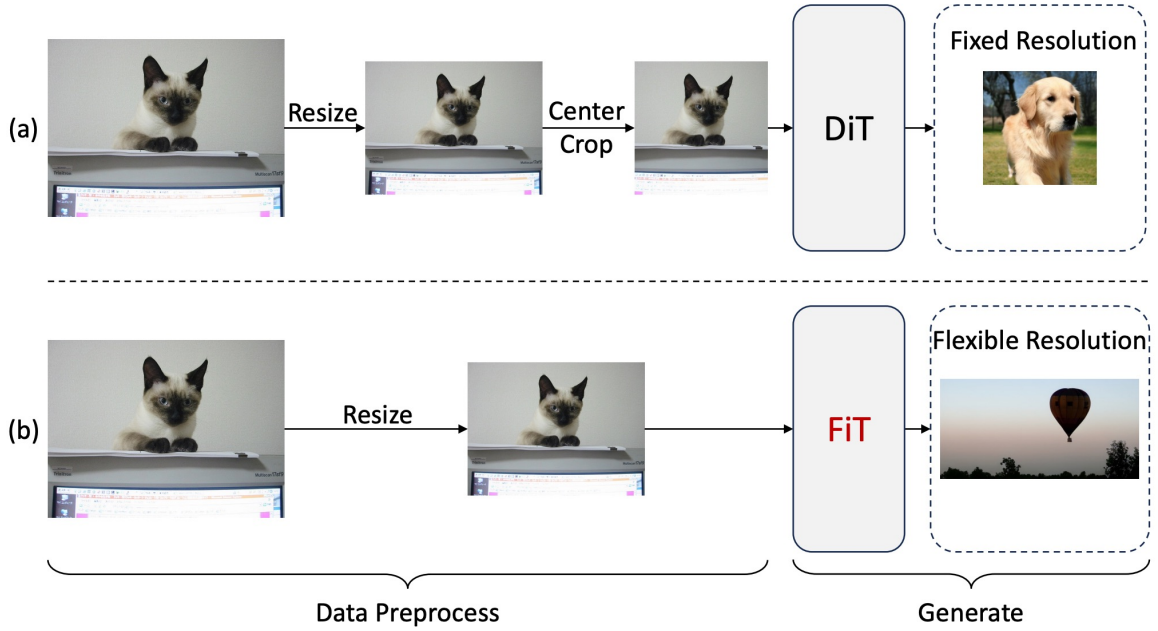
Height/Width distribution of the original ImageNet dataset.

Limitation of DiT

However, DiT struggles with generalizing across arbitrary resolutions.

This limitation stems from the fact that **DiT can not utilize dynamic resolution images** during its training process, hindering its ability to adapt to different token lengths or resolutions effectively.

To overcome this limitation, we introduce the Flexible Vision Transformer (FiT), which is adept at generating images at unrestricted resolutions and aspect ratios.



Pipeline comparison between (a) DiT and (b) FiT.

TL;DR: FiT with Three Updates based on DiT

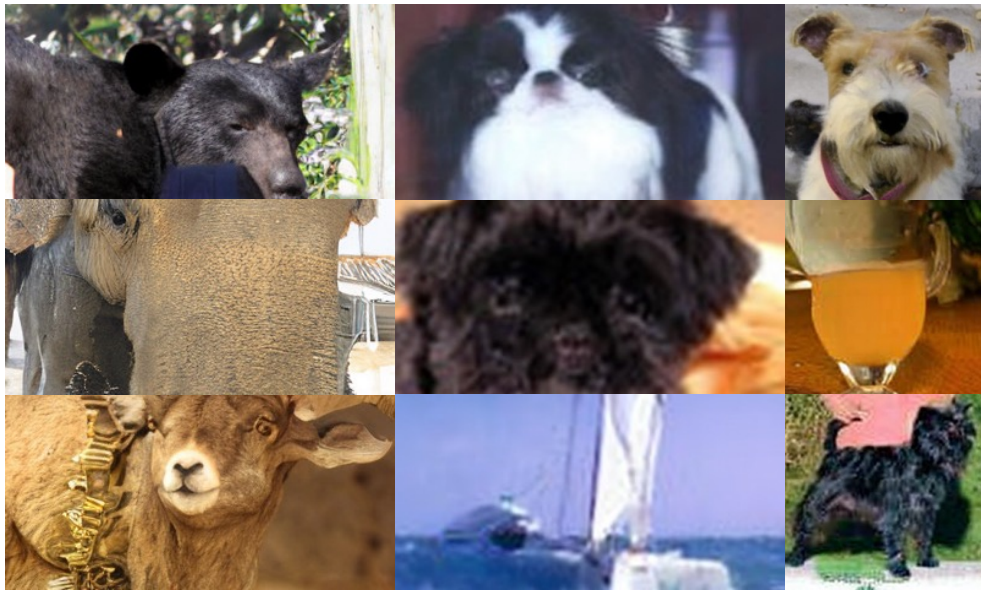
1. Flexible Training and Inference Pipeline
2. Architecture Updates
3. Training Free Resolution Extrapolation Method

Certain Biases in DiT

Due to the diversity in image resolutions, DiT resizes and crops the images to a fixed resolution 256 × 256 for packing the training batch.

While resizing and cropping as a means of data augmentation is a common practice, this approach introduces certain biases into the input data.

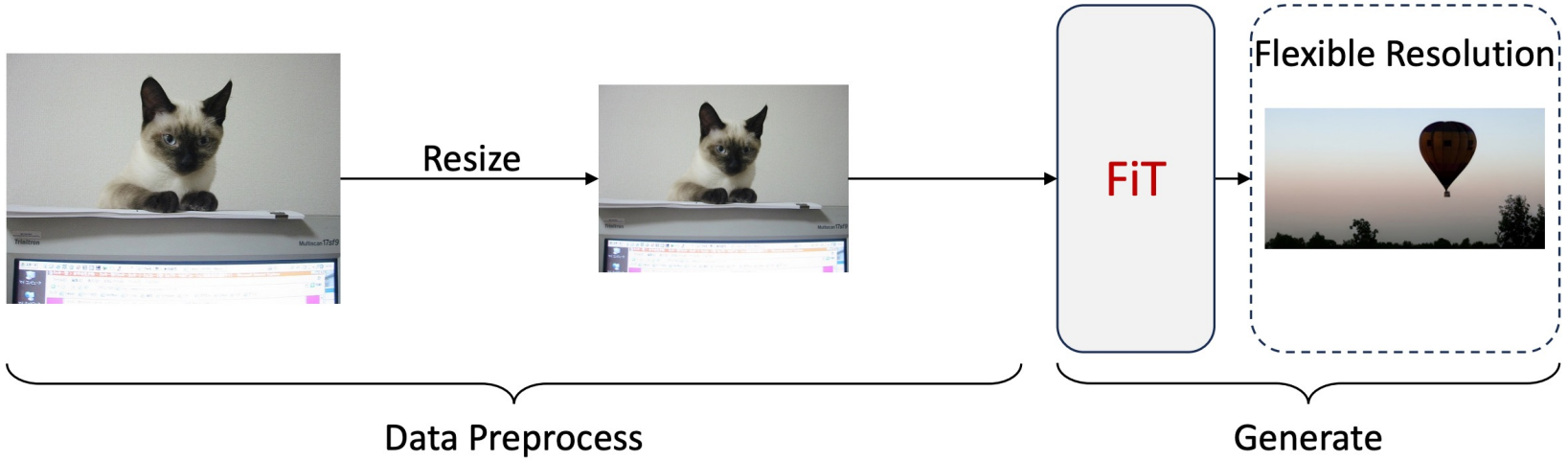
These biases will directly affect the final images generated by the model, including **blurring effects** from the transition from low to high resolution and **information lost due to the cropping**, as shown in right figure.



Failure samples from DiT-XL/2.

Flexible Training and Inference Pipeline

To overcome biases in DiT, we avoid cropping images or resizing low-resolution images to a higher resolution. Instead, we only resize high resolution images to a predetermined maximum resolution limit, $H \times W \leq 256 \times 256$.

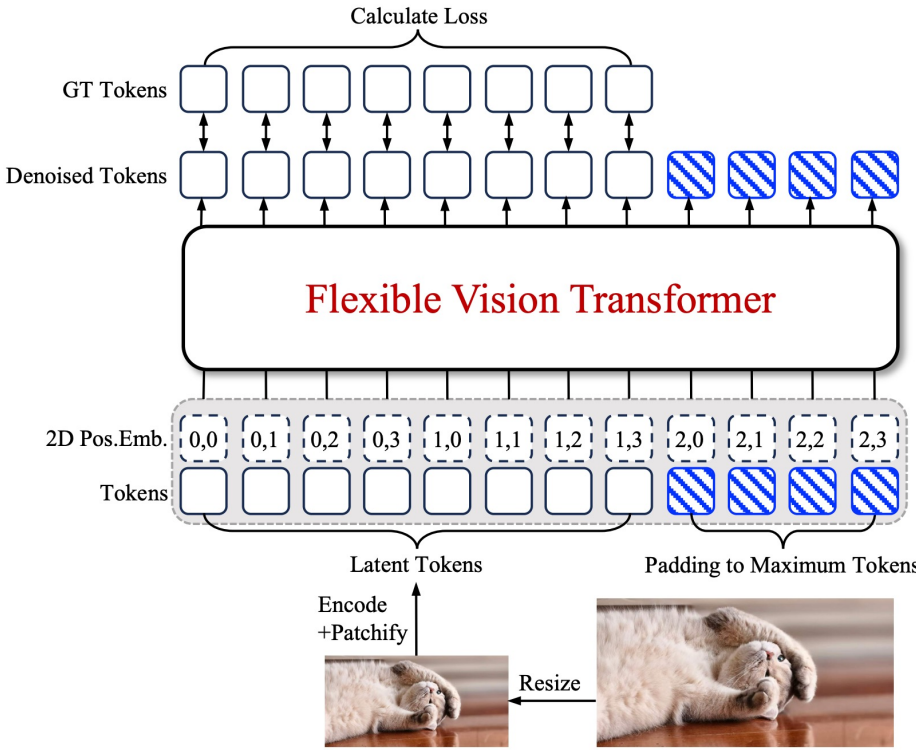


Pipeline of FiT.

Flexible Training Pipeline

Key Idea: *FiT conceptualizes images as sequences of variable length tokens.*

1. FiT first encodes an image into latent codes with a pre-trained VAE encoder. By patchfiying latent codes to latent tokens, we can get sequences with different lengths L.

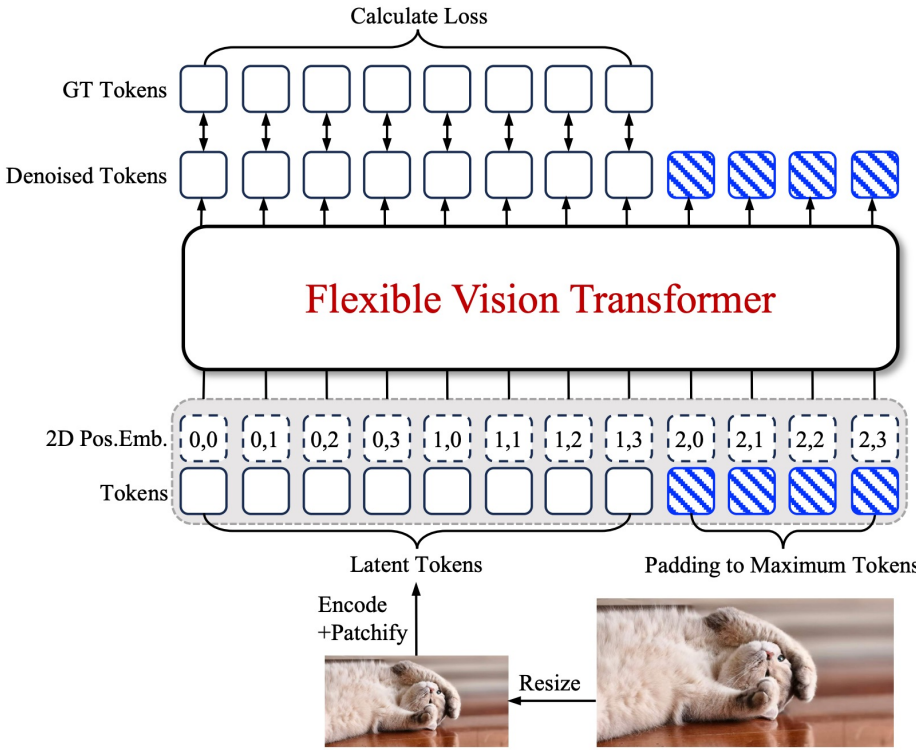


Flexible Training Pipeline.

Flexible Training Pipeline

Key Idea: *FiT conceptualizes images as sequences of variable length tokens.*

2. To pack these sequences into a batch, we pad all these sequences to the maximum token length L_{max} using padding tokens. Here we set $L_{max} = 256$ to match the fixed token length of DiT. The same as the latent tokens, we also pad the positional embeddings to the maximum length for packing.

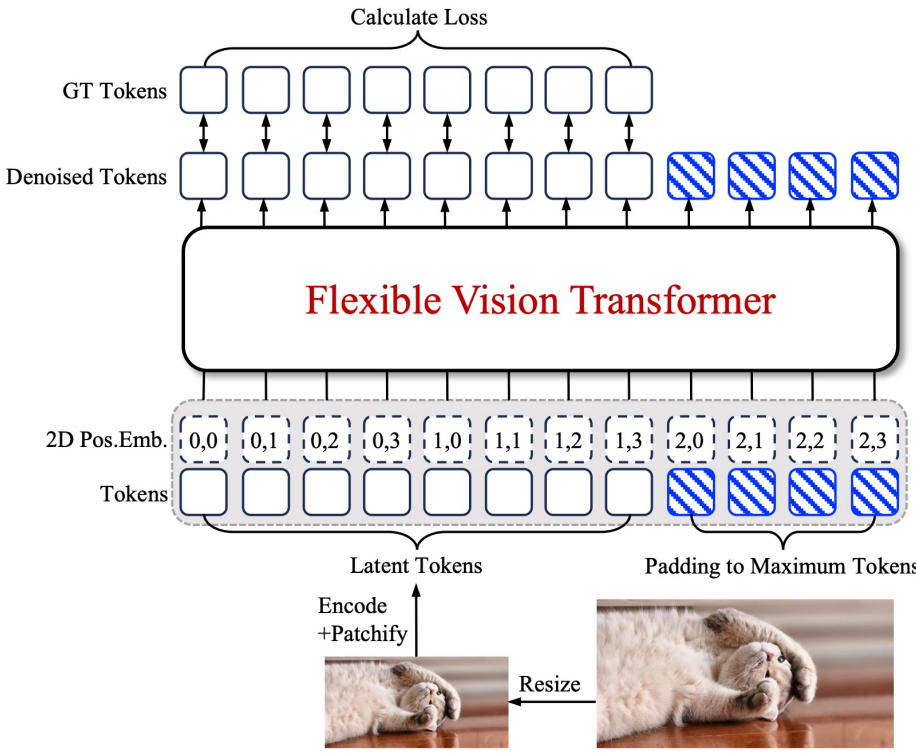


Flexible Training Pipeline.

Flexible Training Pipeline

Key Idea: *FiT conceptualizes images as sequences of variable length tokens.*

3. Finally, we calculate the loss function only for the denoised output tokens, while discarding all other padding tokens.



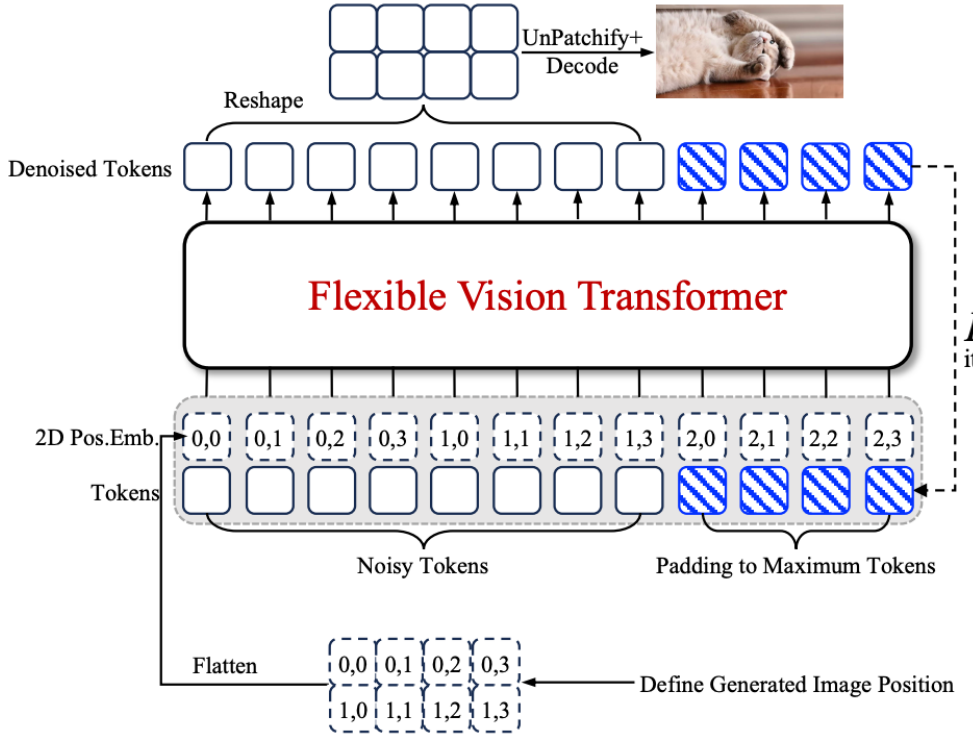
Flexible Training Pipeline.

Flexible Inference Pipeline

Key Idea: *FiT conceptualizes images as sequences of variable length tokens.*

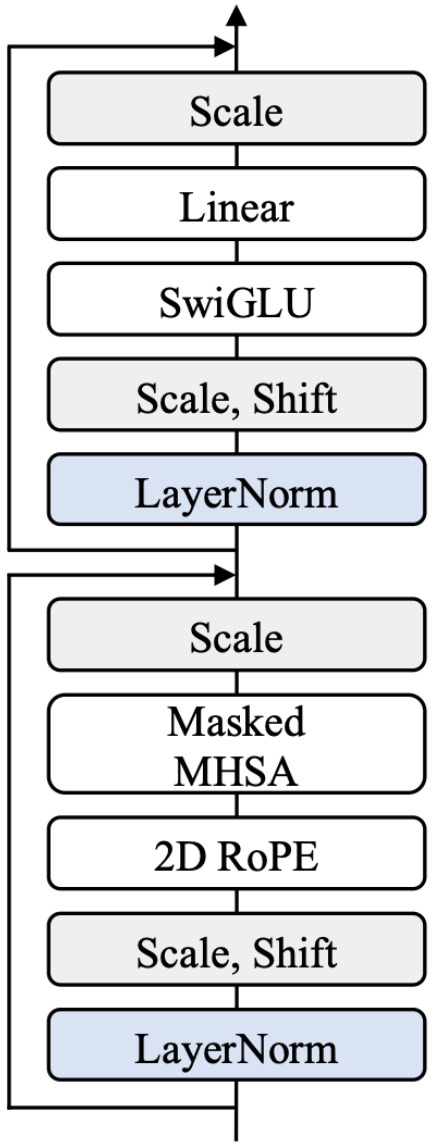
1. We firstly define the position map of the generated image and sample noisy tokens from the Gaussian distribution as input.

2. After completing K iterations of the denoising process, we reshape and unpatchfiy the denoised tokens according to the predefined position map to get the final generated image.



Flexible Inference Pipeline.

FiT Architecture



FiT Block.

Three Updates in FiT Architecture

1. Replacing MHSA with Masked MHSA.

The flexible training pipeline introduces padding tokens for flexibly packing dynamic sequences into a batch. It is crucial to facilitate interactions among noised tokens while **preventing any interaction between noised tokens and padding tokens**.

Here, we utilize the sequence mask M for Masked Attention, where noised tokens are assigned the value of 0, and padding tokens are assigned the value of negative infinity (-inf), which is defined as follows:

$$\text{Masked Attn.}(Q_i, K_i, V_i) = \text{Softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}} + M\right) V_i$$

Three Updates in FiT Architecture

2. Replacing Absolute PE with 2D RoPE.

We observe that vision transformer models with absolute positional embedding **fail to generalize** well on images **beyond the training resolution**.

We utilize 2D-RoPE to facilitate the resolution generalization in vision transformer models.

Three Updates in FiT Architecture

3. Replacing MLP with SwiGLU.

We follow recent LLMs like LLaMA, and replace the MLP in FFN with SwiGLU, which is defined as follows:

$$\mathbf{SwiGLU}(x, W, V) = \mathbf{SiLU}(xW) \otimes (xV)$$

$$\mathbf{FFN}(x) = \mathbf{SwiGLU}(x, W_1, W_2)W_3$$

Training Free Resolution Extrapolation

While large language models employ token length extrapolation techniques (NTK, YaRN) for generating text of arbitrary lengths, a direct application of these technologies to FiT yields suboptimal results.

We tailor these techniques for 2D RoPE, called **VisionNTK** and **VisionYaRN**, thereby enhancing FiT's performance across a spectrum of resolutions and aspect ratios.

It is worth noting that **VisionNTK** and **VisionYaRN** are *training-free positional embedding interpolation approaches*, used to alleviate the problem of position embedding out of distribution in extrapolation.

(When the aspect ratio equals one, they are equivalent to the vanilla implementation of NTK and YaRN. They are especially effective in generating images with arbitrary aspect ratios.)

Architecture Ablation Results

Arch.	Pos. Embed.	FFN	Train	256×256 (i.d.)					160×320 (i.d.)					224×448 (o.o.d.)				
				FID↓	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**	**FID↓**	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**	**FID↓**	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**
DiT-B	Abs. PE	MLP	Fixed	44.83	**8.49**	32.05	0.48	**0.63**	91.32	66.66	14.02	0.21	0.45	109.1	110.71	14.00	0.18	0.31
Config A	Abs. PE	MLP	Flexible	43.34	11.11	32.23	0.48	0.61	50.51	10.36	25.26	0.42	0.60	52.55	16.05	28.69	0.42	**0.58**
Config B	Abs. PE	SwiGLU	Flexible	41.75	11.53	34.55	0.49	0.61	48.66	10.65	26.76	0.41	0.60	52.34	17.73	30.01	0.41	0.57
Config C	Abs. PE + 2D RoPE	MLP	Flexible	39.11	10.79	36.35	0.51	0.61	46.71	10.32	27.65	**0.44**	0.61	46.60	**15.84**	33.99	0.46	**0.58**
Config D	2D RoPE	MLP	Flexible	37.29	10.62	38.34	**0.53**	0.61	45.06	**9.82**	28.87	0.43	0.62	46.16	23.72	35.28	0.46	0.55
FiT-B	2D RoPE	SwiGLU	Flexible	**36.36**	11.08	**40.69**	0.52	0.62	**43.96**	10.26	**30.45**	0.43	**0.62**	**44.67**	24.09	**37.10**	**0.49**	0.53

Ablation results from DiT-B/2 to FiT-B/2 at 400K training steps without using classifier-free guidance.

1. Flexible training pipeline significantly improves the performance across various resolutions.

2. SwiGLU slightly improves the performance across various resolutions, compared to MLP.

3. 2D RoPE demonstrates greater efficiency compared to absolute position encoding, and it possesses significant extrapolation capability across various resolutions.

Resolution Extrapolation Design Ablation Results

Method	320×320 (1:1)					224×448 (1:2)					160×480 (1:3)				
	FID↓	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**	**FID↓**	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**	**FID↓**	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**
DiT-B	95.47	108.68	18.38	0.26	0.40	109.1	110.71	14.00	0.18	0.31	143.8	122.81	8.93	0.073	0.20
DiT-B + EI	81.48	62.25	20.97	0.25	0.47	133.2	72.53	11.11	0.11	0.29	160.4	93.91	7.30	0.054	0.16
DiT-B + PI	72.47	54.02	24.15	0.29	0.49	133.4	70.29	11.73	0.11	0.29	156.5	93.80	7.80	0.058	0.17
FiT-B	61.35	**30.71**	31.01	0.41	0.53	44.67	**24.09**	37.1	0.49	0.52	56.81	**22.07**	25.25	**0.38**	0.49
FiT-B + PI	65.76	65.45	29.32	0.32	0.45	175.42	114.39	8.45	0.14	0.06	224.83	123.45	5.89	0.02	0.06
FiT-B + YaRN	44.76	38.04	44.70	0.51	0.51	82.19	75.48	29.68	0.40	0.29	104.06	72.97	20.76	0.21	0.31
FiT-B + NTK	57.31	31.31	33.97	0.43	0.55	45.24	29.38	38.84	0.47	0.52	59.19	26.54	26.01	0.36	0.49
FiT-B + **VisionYaRN**	**44.76**	38.04	**44.70**	**0.51**	0.51	**41.92**	42.79	**45.87**	**0.50**	0.48	62.84	44.82	**27.84**	0.36	0.42
FiT-B + **VisionNTK**	57.31	31.31	33.97	0.43	**0.55**	43.84	26.25	39.22	0.48	**0.52**	**56.76**	24.18	26.40	0.37	**0.49**

Direct extrapolation does not perform well on larger resolution out of training distribution. So we conduct a comprehensive benchmarking analysis focused on positional embedding interpolation methods.

We find that FiT-B/2 has a strong extrapolation ability, which can be further enhanced when combined with VisionYaRN and VisionNTK methods.

In-distribution Resolution Results

Method	Train Cost	256×256 (1:1)					160×320 (1:2)					128×384 (1:3)				
		FID↓	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**	**FID↓**	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**	**FID↓**	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**
BigGAN-deep	-	6.95	7.36	171.4	0.87	0.28	-	-	-	-	-	-	-	-	-	-
StyleGAN-XL	-	2.30	4.02	265.12	0.78	0.53	-	-	-	-	-	-	-	-	-	-
MaskGIT	1387k×256	6.18	-	182.1	0.80	0.51	-	-	-	-	-	-	-	-	-	-
CDM	-	4.88	-	158.71	-	-	-	-	-	-	-	-	-	-	-	-
U-ViT-H/2-G (cfg=1.4)	500k×1024	2.35	5.68	265.02	0.82	0.57	6.93	12.64	175.08	0.67	0.63	196.84	95.90	7.54	0.06	0.27
ADM-G,U	1980k×256	3.94	6.14	215.84	0.83	0.53	10.26	12.28	126.99	0.67	0.59	56.52	43.21	32.19	0.30	0.50
LDM-4-G (cfg=1.5)	178k×1200	3.60	5.12	247.67	**0.87**	0.48	10.04	11.47	119.56	0.65	0.61	29.67	26.33	57.71	0.44	**0.61**
MDT-G† (cfg=3.8,s=4)	6500k×256	**1.79**	**4.57**	**283.01**	0.81	**0.61**	135.6	73.08	9.35	0.15	0.20	124.9	70.69	13.38	0.13	0.42
DiT-XL/2-G (cfg=1.50)	7000k×256	2.27	4.60	278.24	0.83	0.57	20.14	30.50	97.28	0.49	**0.67**	107.2	68.89	15.48	0.12	0.52
FiT-XL/2-G* (cfg=1.50)	1800k×256	4.27	9.99	249.72	0.84	0.51	**5.74**	**10.05**	**190.14**	**0.74**	0.55	**16.81**	**20.62**	**110.93**	**0.57**	0.52

Benchmarking class-conditional image generation with in-distribution resolution on ImageNet dataset.

FiT-XL/2 outperforms all prior generative models across a spectrum of resolutions and aspect ratios.

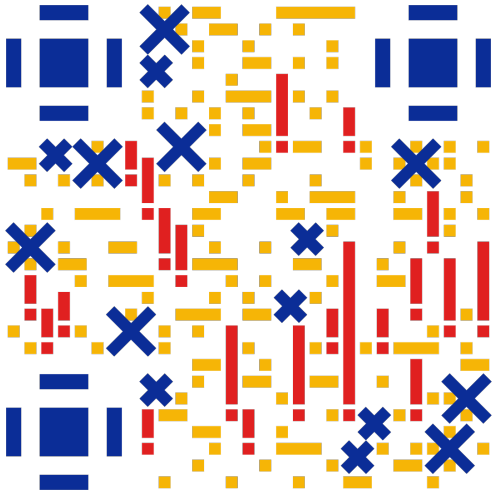
Out-of-distribution Resolution Results

Method	Train Cost	320×320 (1:1)					224×448 (1:2)					160×480 (1:3)				
		FID↓	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**	**FID↓**	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**	**FID↓**	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**
U-ViT-H/2-G (cfg=1.4)	500k×1024	7.65	16.30	208.01	0.72	**0.54**	67.10	42.92	45.54	0.30	**0.49**	95.56	44.45	24.01	0.19	0.47
ADM-G,U	1980k×256	9.39	**9.01**	161.95	0.74	0.50	11.34	**14.50**	146.00	0.71	**0.49**	23.92	25.55	80.73	0.57	**0.51**
LDM-4-G (cfg=1.5)	178k×1200	6.24	13.21	220.03	**0.83**	0.44	8.55	17.62	186.25	**0.78**	0.44	19.24	**20.25**	99.34	0.59	0.50
MDT-G† (cfg=3.8,s=4)	6500k×256	383.5	136.5	4.24	0.01	0.04	365.9	142.8	4.91	0.01	0.05	276.7	138.1	7.20	0.03	0.09
DiT-XL/2-G (cfg=1.50)	7000k×256	9.98	23.57	225.72	0.73	0.48	94.94	56.06	35.75	0.23	0.46	140.2	79.60	14.70	0.094	0.45
FiT-XL/2-G* (cfg=1.50)	1800k×256	**5.42**	15.41	**252.65**	0.81	0.47	**7.90**	19.63	**215.29**	0.75	0.47	**15.72**	22.57	**132.76**	**0.62**	0.47

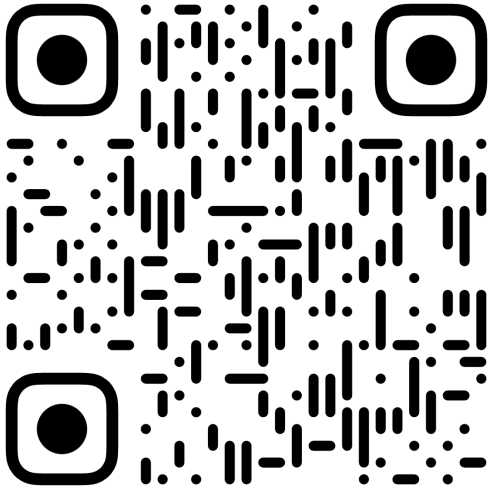
Benchmarking class-conditional image generation with out-of-distribution resolution on ImageNet dataset.

FiT-XL/2 outperforms all prior generative models across a spectrum of resolutions and aspect ratios.

Thank You!



FiT Paper



FiT Github Repo