FiTv2: Scalable and Improved Flexible Vision Transformer for Diffusion Model

ZiDong Wang*, Zeyu Lu*, Di Huang, Cai Zhou, Wanli Ouyang, and Lei Bai †

Abstract—*Nature is infinitely resolution-free*. In the context of this reality, existing diffusion models, such as Diffusion Transformers, often face challenges when processing image resolutions outside of their trained domain. To address this limitation, we conceptualize images as sequences of tokens with dynamic sizes, rather than traditional methods that perceive images as fixed-resolution grids. This perspective enables a flexible training strategy that seamlessly accommodates various aspect ratios during both training and inference, thus promoting resolution generalization and eliminating biases introduced by image cropping. On this basis, we present the **Flexible Vision Transformer** (FiT), a transformer architecture specifically designed for generating images with *unrestricted resolutions and aspect ratios*. We further upgrade the FiT to FiTv2 with several innovative designs, includingthe Query-Key vector normalization, the AdaLN-LoRA module, a rectified flow scheduler, and a Logit-Normal sampler. Enhanced by a meticulously adjusted network structure, FiTv2 exhibits $2\times$ convergence speed of FiT. When incorporating advanced training-free extrapolation techniques, FiTv2 demonstrates remarkable adaptability in both resolution extrapolation and diverse resolution generation. Additionally, our exploration of the scalability of the FiTv2 model reveals that larger models exhibit better computational efficiency. Furthermore, we introduce an efficient post-training strategy to adapt a pre-trained model for the high-resolution generation. Comprehensive experiments demonstrate the exceptional performance of FiTv2 across a broad range of resolutions. We have released all the codes and models at https://github.com/whlzy/FiT to promote the exploration of diffusion transformer models for arbitrary-resolution image generation.

Index Terms—Vision Transformer, Diffusion Model.

1 Introduction

Natural images inherently possess various resolutions, as illustrated in Fig. 3, the images in *ImageNet* [1] showcase diverse resolutions and aspect ratios. However, current image generation models struggle with generalizing across arbitrary resolutions. The Diffusion Transformer (DiT) [2] family, while excelling within certain resolution ranges, falls short when dealing with images of varying resolutions. This constraint arises from the inability of DiT to incorporate images with dynamic resolutions during its training process, thereby impeding its capability to adapt to diverse token lengths or resolutions effectively.

To bridge this gap, we introduce the **Flexible Vision Transformer (FiT)**, a novel architecture adept at generating images at *unrestricted resolutions and aspect ratios*. The core motivation lies in a fundamental shift in image data conceptualization:: FiT conceptualizes images as sequences of variable-length tokens, departing from the traditional perspective of static grids with fixed dimensions. This paradigm enables FiT to dynamically adjust the sequence length, thereby facilitating the generation of images at any desired resolution without being constrained by pre-defined image grids. By efficiently managing variable-length token sequences and padding them to a maximum specified length, FiT unlocks the potential for resolution-independent image synthesis. FiT represents this paradigm shift through significant advancements in the **flexible training pipeline**, the **network architecture**, and the **inference process**.

Flexible Training Pipeline. FiT introduces a unique approach by preserving the original image aspect ratio during training, conceptualizing each image as a sequence of tokens. This innovative perspective enables FiT to adaptively resize high-resolution images to conform to a predefined maximum token limit, ensuring that no image, regardless of its original dimensions, undergoes cropping or disproportionate scaling. As illustrated in Fig. 2, this methodology maintains the integrity of the image resolution, facilitating the generation of high-fidelity images across a diverse range of resolutions. To the best of our knowledge, FiT is the first transformer-based generation model capable of accommodating varied image resolutions throughout training.

Network Architecture. The FiT model evolves from the DiT [3] architecture, addressing its limitations in resolution extrapolation. The most essential network architecture adjustment to handle diverse image sizes is the adoption of 2-D Rotary Positional Embedding (2-D RoPE) [4], inspired by its success in enhancing large language models (LLMs) for length extrapolation [5]. Additionally, we introduce Swish-Gated Linear Unit (SwiGLU) [6] in place of the traditional Multi-layer Perceptron (MLP) and replace the Multi-Head Self-Attention (MHSA) of DiT with Masked MHSA to effi-

- *: *ZiDong Wang and Zeyu Lu contribute equally to this project.*
- *Zidong Wang, Zeyu Lu, Wanli Ouyang, and Lei Bai are with the Shanghai AI Laboratory, Shanghai, 200000, China.*
- *Zidong Wang and Wanli Ouyang are with the Chinese University of Hong Kong, Shatin, 999077, Hong Kong.*
- *Zeyu Lu is with the Shanghai Jiao Tong University, Shanghai, 200000, China.*
- *Di Huang is with the University of Sydney, Camperdown NSW 2050, Australia.*
- *Cai Zhou is with the Tsinghua University, Beijing, 100084, China.*
- †: *Corresponding author is Lei Bai. Email: baisanshi@gmail.com.*

Fig. 1: **Selected samples from FiTv2-3B/2 models at resolutions of** 256×256**,** 512×512**,** 768×768**,** 256×768 **and** 768×256**.** All the images are sampeld with CFG=4.0. FiT is capable of generating images at unrestricted resolutions and aspect ratios. FiTv2 pushes the image generation ability of FiT to a new level, capable of generating better and higher-resolution images.

ciently manage padding tokens within our flexible training pipeline. It is noteworthy that we incorporate several advanced designs into our FiTv2, such as the Adaptive Layer Normalization with Low-Rank Adaptation [7] (AdaLN-LoRA), further imporving both efficiency and scalability.

Inference Process. While large language models employ token length extrapolation techniques [8], [9] for generating text of arbitrary lengths, the direct application of these technologies to FiT yields suboptimal results. We tailor these techniques for 2-D RoPE, enhancing FiT's performance across a spectrum of resolutions and aspect ratios.

To achieve better performance, we adopt several advanced enhancements to build FiTv2, an improved version of FiT. Extensive experiments on both class-guided image generation and text-to-image generation tasks demonstrate that our FiTv2 outperforms or achieves competitive performance compared to other state-of-the-art CNN [10], [11] and transformer models [2], [12]. Specifically, our FiTv2-3B/2 model, after training only $1000K$ steps on *ImageNet* [1] dataset, achieves competitive performance on standard *ImageNet*-256 $\times$ 256 benchmark while outperforming all SOTA models by a significant margin across resolutions of 160×320, 128×384, 320×320, 224×448, and 160×480. With merely $200K$ extra post-training steps, our FiTv2-3B/2 model exceeds all SOTA models by a great margin across 512×512, 320×640, and 256×768 resolutions. Further, with the same training steps, our FiTv2-XL/2 model surpasses the SiT [12]-XL/2 model greatly on text-to-image tasks.

A preliminary version of this work was published in [13]. In this paper, we extend the conference version in the following aspects:

- We propose an improved version of FiT by incorporating Query-Key Vector Normalization (QK-Norm) into the attention layer for stability, as well as decreasing the hidden size of Swish-Gated Linear Unit (SwiGLU) [6] and adopting the Adaptive Layer Normalization with Low-Rank Adaptation [7] (AdaLN-LoRA) for efficiency. These improvements lead to FiTv2, a more efficient and scalable version of FiT, which achieves state-of-the-art performance in many image generation tasks.
- We improve the training strategy by switching the denoising model from denoising diffusion probabilistic model (DDPM) [14] to rectified flow [15] and adopting the Logit-Normal sampling for timesteps, which results in faster convergence. Furthermore, we analyze the limitations of the original FiT and propose a novel mixed data preprocessing strategy that benefits image synthesis across various resolutions. Combining the above architectural and training strategy improvement enables FiTv2 to achieve 2× the convergence speed of the original FiT.
- We provide comprehensive analytical experiments and visualization to evaluate the effectiveness of FiT and FiTv2. In addition to the study of FiT core components in Sec. 5.2, we comprehensively analyze the effect of each modification from FiT to FiTv2, as detailed in Sec. 5.3. We explore different training-free resolution extrapolation methods for arbitrary resolution generation in FiTv2. Moreover, we benchmark the generalization and extrapolation performance of FiTv2 and other state-of-the-art methods. We also scale our FiTv2 model to 3 billion parameters to study the scalability. Furthermore, we conduct an efficient post-training experiment to investigate the transfer from low resolution to high resolution. To validate the effectiveness of FiTv2 beyond the class-guided image generation, we extend it to text-to-image generation tasks, demonstrating its superiority over the previous state-of-the-art SiT [12] model.

2 Related Works

2.1 Diffusion Model

Denoising diffusion probabilistic models (DDPMs) [14], [16], [17], [18], [19] and score-based generative models [20], [21] have exhibited remarkable progress in the context of image generation tasks [3], [10], [16], [22], [23], [24]. The Denoising Diffusion Implicit Model (DDIM) [25], offers an accelerated sampling procedure. Latent Diffusion Models (LDMs) [10] establishes a new benchmark of training deep generative models to reverse a noise process in the latent space, through the use of VAE [26]. Rectified flow model [15] learns a neural ordinary differential equation (ODE) that transports between two distributions. By solving a nonlinear least squares optimization problem, rectified flow learns to map the points drawn from two distributions following the straight paths, which are the shortest paths between two points and hence yield computational efficiency. We follow the rectified flow implementation in SiT [12] for image synthesis with fewer sampling steps.

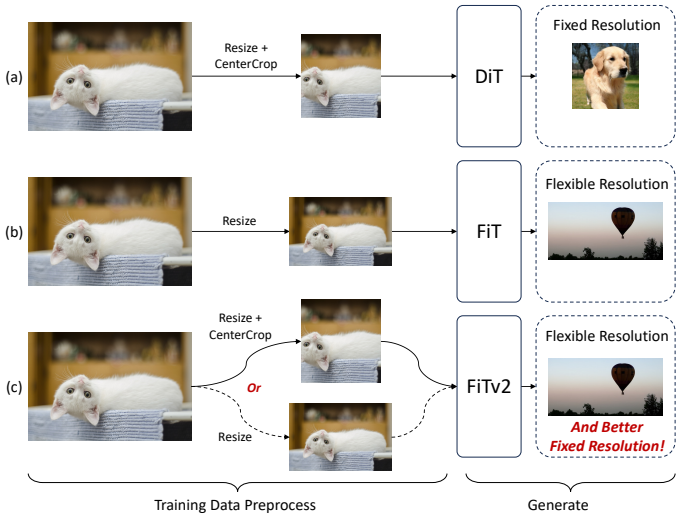


Fig. 2: **Pipeline comparison between (a) DiT, (b) FiT, and (c) FiTv2.** In FiTv2, we incorporate both fixed-resolution images and the flexible-resolution images into our training process.

2.2 Diffusion Transformer

The Transformer model [27], has successfully supplanted domain-specific architectures in a variety of fields including, but not limited to, language [28], [29], vision [30], [31], and multi-modality [32]. In vision perception research, most efforts [33], [34], [35], [36] are aimed at accelerating pretraining using a fixed, low resolution. On the other hand, NaViT [37] implements the *Patch n' Pack* technique to train vistion transformer (ViT) using images at their native aspect ratios. Notably, transformers have been also explored in the denoising diffusion probabilistic models [14] to synthesize images. DiT [2] is the seminal work that utilizes a vision transformer as the backbone of LDMs and can serve as a strong baseline. Based on DiT architecture, MDT [38] introduces a masked latent modeling approach, which requires two forward runs in training and inference. U-ViT [39] treats all inputs as tokens and incorporates U-Net architectures into the ViT backbone of LDMs. DiffiT [40] introduces a time-dependent self-attention module into the DiT backbone to adapt to different stages of the diffusion process. SiT [12] utilizes the same architecture as DiT and explores different rectified flow configurations. Large-DiT and Flag-DiT [41] scale up the diffusion transformers and achieve better performance. We follow the LDM paradigm of the above methods and further propose a novel flexible image synthesis pipeline.

2.3 Length Extrapolation in LLM

Rotary Position Embedding (RoPE) [4] is a pivotal advancement in positional embedding techniques for large language models (LLM). By applying a rotary transformation to the embeddings, it incorporates relative position information into absolute positiaonal embedding. This prowerful representation capability has made it the dominant positional embedding in a wide range of LLM designs [42], [43], [44], [45], [46]. Although RoPE enjoys valuable properties, such as the flexibility of sequence length, its performance drops when

the input sequence surpasses the training length. Many approaches have been proposed to solve this issue. Position Interpolation (PI) [47] linearly down-scales the input position indices to match the original context window size, while NTK-aware Scaled RoPE Interoplation [9] changes the rotary base of RoPE based on the Neural Tangent Kernel (NTK) theory. YaRN (Yet another RoPE extensioN) [8] is an improved method to efficiently extend the context window. RandomPE [48] sub-samples an ordered set of positions from a much larger range of positions than originally observed in training or inference. xPos [49] incorporates long-term decay into RoPE and uses blockwise causal attention for better extrapolation performance. While these methods scale the the positional embedding to accommodate longer contexts during inference, another paradigm [50], [51] directly scales the attention logits to aggergate information based on entropy theory. Our work delves deeply into the implementation of RoPE in vision generation and provides a comprehensive benchmark for diverse methods on image resolution extrapolation and generalization.

3 PRELIMINARIES

3.1 Diffusion Model

DDPM. Denoising diffusion probabilistic models (DDPMs) [14] seek to learn a model distribution $p_\theta(\mathbf{x}_0)$ that closely approximates the ground truth data distribution $q(\mathbf{x}_0)$ and allows for easy sampling:

$$p_\theta(\mathbf{x}_0) = \int p_\theta(\mathbf{x}_{0:T})\mathrm{d}\mathbf{x}_{1:T} \tag{1}$$

where

$$p_\theta(\mathbf{x}_{0:T}) := p_\theta(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta^{(t)}(\mathbf{x}_{t-1}|\mathbf{x}_t) \tag{2}$$

The forward process or diffusion process, represented by the approximate posterior $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$, is a fixed Markov chain that progressively adds Gaussian noise to the data:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{3}$$

where

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta\mathbf{I}) \tag{4}$$

The forward process variance schedule $\beta_1, \ldots, \beta_T$ can be set as hyperparameters or learned through reparameterization [14]. This parameterization leads to a closed-form expression of conditional probability:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}) \tag{5}$$

where $\alpha_t := 1-\beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t}\alpha_s$. Using the closed-form expression, we can express $\mathbf{x}_t$ as a linear combination of $\mathbf{x}_0$ and a noise variable $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \tag{6}$$

Note that $\bar{\alpha}_t$ is a decreasing sequence, and when $\bar{\alpha}_T$ is set sufficiently close to 0, $q(\mathbf{x}_T|\mathbf{x}_0)$ converges to a standard Gaussian for all $\mathbf{x}_0$, allowing us to set $p_\theta(\mathbf{x}_T) := \mathcal{N}(\mathbf{0}, \mathbf{I})$.

In the generative process or the reverse process, the model is trained to fit the data distribution $q(\mathbf{x}_0)$ by approximating the intractable reverse process $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ through maximizing a variational lower bound:

$$\begin{aligned}&\max_\theta \mathbb{E}_{q(\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_0)]\\ \le &\max_\theta \mathbb{E}_{q(\mathbf{x}_0,\mathbf{x}_1,\ldots,\mathbf{x}_T)}[\log p_\theta(\mathbf{x}_{0:T}) - \log q(\mathbf{x}_{1:T}|\mathbf{x}_0)]\end{aligned} \tag{7}$$

If all conditionals $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ are modeled as Gaussians with trainable mean functions and fixed variances as in [14], the objective in Eq. (7) can be simplified to:

$$\mathcal{L}_\gamma(\epsilon_\theta) := \sum_{t=1}^{T}\lambda_t \mathop{\mathbb{E}}_{\substack{\mathbf{x}_0\sim q(\mathbf{x}_0)\\ \epsilon_t\sim\mathcal{N}(\mathbf{0},\mathbf{I})}}\left[||\epsilon_\theta^{(t)}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t) - \epsilon_t||_2^2\right] \tag{8}$$

where $\epsilon_\theta := \{\epsilon_\theta^{(t)}\}_{t=1}^T$ is a set of T functions, each $\epsilon_\theta^{(t)}$ indexed by t is a denoising function with trainable parameters $\theta^{(t)}$; $\lambda := [\lambda_1, \ldots, \lambda_T]$ is a set of positive coefficients, set to all ones in [14].

In the inference stage for a trained model, $\mathbf{x}_0$ is sampled by first sampling $\mathbf{x}_T$ from the prior $p_\theta(\mathbf{x}_T)$, followed by sampling $\mathbf{x}_{t-1}$ from the above generative processes iteratively. The length of steps T in the forward process is a hyperparameter in DDPMs. Larger T enables the Gaussian conditionals distributions as better approximations, at the cost of making the sequential sampling slower.

DDPM and score-based generative model can both be formulated through the system of stochastic differential equations (SDE) [21], which are able to generate high-quality samples yet slow in the inference stage due to the iterative denoising process. [25] proposes DDIM, an implicit probabilistic model that samples with a fixed procedure corresponding to ordinary differential equations (ODE). The ODE-based acceleration leads to fewer discretization steps in sampling but tends to have lower generation quality compared with SDE-based methods.

Rectified flow. To tackle the aforementioned problem, [15] proposes rectified flow, an ODE-based model that transports two empirical distributions π_0 to π_1 by following straight line paths as much as possible. The straight paths are both theoretically desired since they are the shortest paths between two endpoints, and computationally efficient because they can be simulated exactly without time discretization, allowing for few-step and even one-step sampling.

Given two target distributions π_0, π_1 and empirical observations $X_0 \sim \pi_0, X_1 \sim \pi_1$, the rectified flow induced from (X_0, X_1) is an ODE model on time $t \in [0, 1]$,

$$\mathrm{d}Z_t = v(Z_t, t)\mathrm{d}t \tag{9}$$

which converts Z_0 from π_0 to Z_1 from π_1. Here the drift force $v : \mathbb{R}^d \to \mathbb{R}^d$ aims to drive the flow to follow the straight direction $(X_1 - X_0)$ as much as possible. To learn this force following the linear path pointing from X_0 to X_1, only a simple least squares regression problem needs to be solved:

$$\min_v \int_0^1 \mathbb{E}\Big[||(X_1 - X_0) - v(X_t, t)||^2\Big]\mathrm{d}t \tag{10}$$

where $X_t := tX_1 + (1-t)X_0$ is the linear interpolation of X_0 and X_1. In practice, v is parameterized with neural networks.

Rectified flow yields several desired properties. First, the flows avoid crossing different paths, which is the condition that the ODE is well-defined, i.e., its solution exists and is unique. While Z_t causalizes, Markovianizes, and derandomizes X_t, it preserves the marginal distributions all the time because the continuity equation always holds. Theoretically, rectified flow provably reduces the convex transport costs: $\mathbb{E}[c(Z_1 - Z_0)] \leq \mathbb{E}[c(X_1 - X_0)]$ for any convex $c : \mathbb{R}^d \to \mathbb{R}$. An intuitive explanation is that the paths of the flow Z_t is a rewiring of the straight paths connecting (X_0, X_1), thus the convex transport costs are guaranteed to decrease. Furthermore, on the practical computational efficiency side, the flow becomes nearly straight with just one step of reflow, hence a very few number of Euler discretization steps or even a single Euler step is needed to simulate the ODE. This not only reduces discretization error but also largely improves the sample efficiency.

3.2 Rotary Positional Embedding

1-D RoPE (Rotary Positional Embedding). 1-D RoPE [4] is a type of position embedding that unifies absolute and relative PE, exhibiting a certain degree of extrapolation capability in LLMs. Given the m-th key and n-th query vector as $\mathbf{q}_m, \mathbf{k}_n \in \mathbb{R}^{|D|}$, 1-D RoPE multiplies the bias to the key and query vector in the complex vector space:

$$f_q(\mathbf{q}_m, m) = e^{im\Theta}\mathbf{q}_m, \quad f_k(\mathbf{k}_n, n) = e^{in\Theta}\mathbf{k}_n \tag{11}$$

where $\Theta = \mathrm{Diag}(\theta_1, \cdots, \theta_{|D|/2})$ is rotary frequency matrix with $\theta_d = b^{-2d/|D|}$ and rotary base $b = 10000$. In the real space, given $l = |D|/2$, the rotary matrix $e^{im\Theta}$ equals to:

$$\begin{bmatrix} \cos m\theta_1 & -\sin m\theta_1 & \cdots & 0 & 0 \\ \sin m\theta_1 & \cos m\theta_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \cos m\theta_l & -\sin m\theta_l \\ 0 & 0 & \cdots & \sin m\theta_l & \cos m\theta_l \end{bmatrix} \tag{12}$$

The attention score with 1-D RoPE is calculated as:

$$A_n = \mathrm{Re}\langle f_q(\mathbf{q}_m, m), f_k(\mathbf{k}_n, n)\rangle \tag{13}$$

NTK-aware Scaled RoPE Interpolation. It is a training-free length extrapolation technique used in LLMs [9]. To handle larger context length L_{test} than maximum training length L_{train}, it modifies the rotary base of 1-D RoPE as follows:

$$b' = b \cdot s^{\frac{|D|}{|D|-2}}, \tag{14}$$

where the scale factor s is defined as:

$$s = \max(\frac{L_{test}}{L_{train}}, 1.0). \tag{15}$$

YaRN (Yet another RoPE extensioN) Interpolation. [8] introduces the ratio of dimension d as $r(d) = L_{\text{train}}/(2\pi b^{2d/|D|})$, and modifies the rotary frequency as:

$$\theta'_d = (1 - \gamma(r(d)))\,\frac{\theta_d}{s} + \gamma(r(d))\theta_d, \tag{16}$$

where s is the aforementioned scale factor, and $\gamma(r(d))$ is a ramp function with extra hyper-parameters α, β:

$$\gamma(r) = \begin{cases} 0, & \text{if } r < \alpha \\ 1, & \text{if } r > \beta \\ \frac{r-\alpha}{\beta-\alpha}, & \text{otherwise.} \end{cases} \tag{17}$$

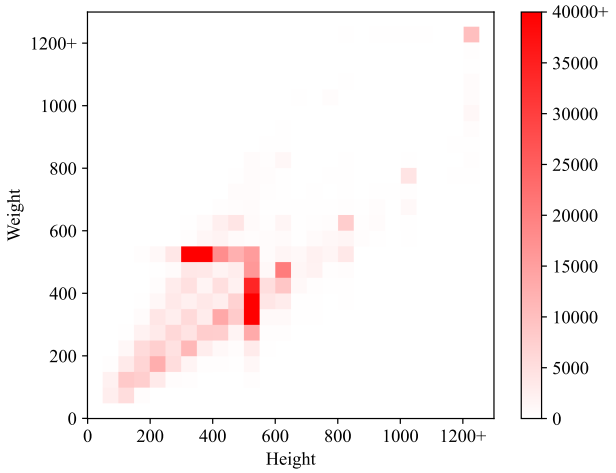


Fig. 3: The Height/Width distribution of the original *ImageNet* [1] dataset.

In addition, it incorporates a 1-D RoPE scaling term as:

$$f'_q(\mathbf{q}_m, m) = \frac{1}{\sqrt{t}} f_q(\mathbf{q}_m, m), f'_k(\mathbf{k}_n, n) = \frac{1}{\sqrt{t}} f_k(\mathbf{k}_n, n), \tag{18}$$

where $\frac{1}{\sqrt{t}} = 0.1\ln(s) + 1$.

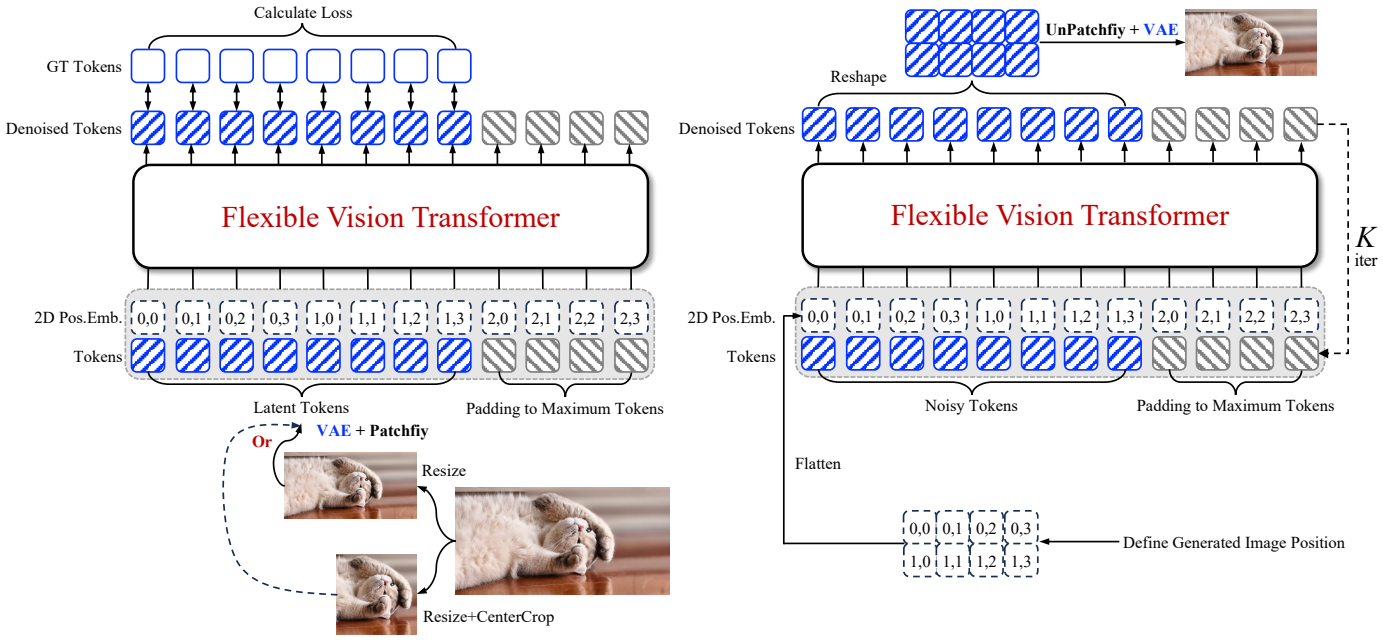
4 Flexible Vision Transformer

4.1 Flexible Training and Inference Pipeline

Modern deep learning models, constrained by the limitations of GPU hardware, are required to pack data into batches of uniform shape for parallel processing. This requirement poses challenges when dealing with the diverse image resolutions prevalent in datasets, as exemplified in Fig. 3. To address this dilemma, DiT [2] simply resizes and crops the images to a fixed resolution 256×256, as in Fig. 4 (a). While this method of data augmentation is widely adopted, it introduces inherent biases into the input data. These biases will directly affect the final images generated by the model, including blurring effects introduced by the transition from low to high resolution and information loss due to the cropping (additional failure samples of DiT can be found in supplementary material). To this end, we propose a flexible training and inference pipeline to solve this challenge elegantly, as illustrated in Fig. 4 (b, c).

In the preprocessing phase, we avoid resizing low-resolution images to a higher resolution to avoid blurring issues. For the original FiT, we only resize high-resolution images to a predetermined maximum resolution limit, $HW \leqslant 256^2$, as in Fig. 2 (b). For FiTv2, we introduce a bifurcated preprocessing pipeline, as shown in Fig. 2 (c). This enhanced approach incorporates both resizing and standard cropping operations into 50% of the preprocessing steps, while the remaining 50% adhere to the original FiT preprocessing pipeline, as detailed in Sec. 4.3.2.

In the training phase, FiT first encodes the preprocessed images into image latents with a pre-trained VAE encoder. Then image latents are patchified to latent tokens to get sequences with different lengths L. 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.



(a) Flexible Training Pipeline (b) Flexible Inference Pipeline

Fig. 4: **Overview of (a) flexible training pipeline, and (b) flexible inference pipeline.** We conceptualize images as dynamic sequences of tokens, allowing for flexible image generation across different resolutions and aspect ratios.

Analogous to the latent tokens, we also pad the positional embeddings to the maximum length for packing. Finally, we calculate the loss function only for the denoised output tokens, discarding all other padding tokens.

In the inference phase, we first define the position map of the generated image and sample noisy tokens from the Gaussian distribution as input. After completing K iterations of the denoising process, we reshape and unpatchify the denoised tokens according to the predefined position map to get the final generated image.

4.2 Flexible Vision Transformer Architecture

Leveraging our flexible training pipeline, our goal is to develop an architecture that can stably train across various resolutions and generate images with arbitrary resolutions and aspect ratios, as shown in Fig. 5 (a). Motivated by some significant architectural advances in LLMs, we conduct a series of experiments to explore architectural modifications based on DiT, see details in Sec. 5.2.

Replacing MHSA with Masked MHSA. The flexible training pipeline introduces padding tokens to accommodate dynamic sequences within batches. To preserve the integrity of the learning process, it's essential to facilitate interactions among noised tokens while isolating them from padding tokens during the the forward phase of transformer. The standard Multi-Head Self-Attention (MHSA) in DiT lacks this capability to distinguish noised tokens and padding tokens. To this end, we use Masked MHSA to replace the standard MHSA. We utilize the sequence mask M for Masked Attention, where the noised tokens are assigned the value of 0, and padding tokens are assigned the value of negative infinity (*-inf*), as defined below:

$$\text{MaskedAttn}_i = \text{Softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}} + M\right) V_i, \qquad (19)$$

where Q_i, K_i, and V_i are the query, key, and value matrices for the i-th head.

Replacing absolute PE with 2-D RoPE. Our experiments in Secs. 5.4 and 5.5.2, reveal that diffusion transformer models employing absolute positional embedding struggle to generalize well beyond their training resolution. Inspired by the success of 1-D RoPE in LLMs for length extrapolation [5], we implement the 2-D RoPE to enhance the resolution generalization in diffusion transformer models. Formally, we calculate the 1-D RoPE separately for height and width coordinates. Then such two 1-D RoPEs are concatenated in the last dimension. Given 2-D coordinates of width and height as $\{(w,h)\big|1 \leqslant w \leqslant W, 1 \leqslant h \leqslant H\}$, the 2-D RoPE is:

$$\begin{aligned} f_q(\mathbf{q}_m, h_m, w_m) &= [e^{ih_m\Theta}\mathbf{q}_m \parallel e^{iw_m\Theta}\mathbf{q}_m], \\ f_k(\mathbf{k}_n, h_n, w_n) &= [e^{ih_n\Theta}\mathbf{k}_n \parallel e^{iw_n\Theta}\mathbf{k}_n], \end{aligned} \qquad (20)$$

where $\Theta = \text{Diag}(\theta_1, \cdots, \theta_{|D|/4})$, and $\parallel$ denotes concatenating two vectors in the last dimension. Note that we divide the $|D|$-dimension space into $|D|/4$-dimension subspace to ensure the consistency of dimension, which differs from $|D|/2$-dimension subspace in 1-D RoPE. Analogously, the attention score with 2-D RoPE is:

$$A_n = \text{Re}\langle f_q(\mathbf{q}_m, h_m, w_m), f_k(\mathbf{k}_n, h_n, w_n)\rangle. \qquad (21)$$

It is noteworthy that there is no cross-term between h and w in 2-D RoPE and attention score A_n, so we can further decouple the rotary frequency as Θ_h and Θ_w, resulting in the decoupled 2-D RoPE, which will be discussed in Sec. 4.4 and more details can be found in supplementary material.

Replacing MLP with SwiGLU. Following recent LLMs like LLaMA [43], [44], we replace the MLP in Feed-forward Neural Network (FFN) with SwiGLU, which is defined as:

$$\begin{aligned} \text{SwiGLU}(x, W, V) &= \text{SiLU}(xW) \otimes (xV) \\ \text{FFN}(x) &= \text{SwiGLU}(x, W_1, W_2)W_3 \end{aligned} \qquad (22)$$

where $\otimes$ denotes Hadmard Product, W_1, W_2, and W_3 are the weight matrices without bias, $\text{SiLU}(x) = x \otimes \sigma(x)$. Here we will use SwiGLU as our choice in each FFN block.

4.3 Enhancing FiT to FiTv2

4.3.1 Architecture

We conduct extensive experiments to further improve the design of FiT blocks that enable more stable and efficient training, as detailed in Sec. 5.3. The architecture changes from FiT to FiTv2 block are illustrated in Fig. 5.

Adding QK-Norm to stabilize training. We observe a vanishing loss problem when scaling up the training steps of the original FiT under mixed-precision training, as in Tab. 3. Inspired by the ViT-22B [52], we apply LayerNorm (LN) to the Query (Q) and Key (K) vectors before the attention calculation. Formally, the attention weights in Eq. (19) is modified to:

$$\text{Softmax}(\frac{1}{\sqrt{d_k}}\text{LN}(Q_i)\text{LN}(K_i)^T + M). \tag{23}$$

By applying this technique, we can effectively eliminate excessively large values in attention logits, which stabilizes the training process, particularly during our mixed-precision training.

Reassigning model parameters. We find that directly using SwiGLU with the same hidden size as the original MLP in DiT [3] will incur more parameters and computational cost, as detailed in Tab. 1. To align the parameters and FLOPs with the baseline (the MLP in DiT), the hidden size of SwiGLU in FiTv2 is set to $\frac{2}{3}\times$ of that in the original FiT.

Given the hidden size as d, the main parameters of a FiT block are composed of:

$$N = N_{\text{attn}} + N_{\text{swiglu}} + N_{\text{AdaLN}} = 4 \cdot d^2 + 8 \cdot d^2 + 6 \cdot d^2 \tag{24}$$

The parameter ratio of Attention, SwiGLU, and AdaLN module is $2:4:3$. We argue that too many parameters are occupied by the AdaLN module, which reduces the capacity available for self-attention blocks and potentially affects the scalability of the model. Inspired by W.A.L.T. [53], we adopt AdaLN-LoRA in our FiTv2 block. Additionally, a global AdaLN module is utilized to capture overlapping condition information and reduce the redudancy of condition information of each block. This global AdaLN module is shared by N blocks, as shown in Fig. 5.

Let $S^i = [\beta_1^i, \beta_2^i, \gamma_1^i, \gamma_2^i, \alpha_1^i, \alpha_2^i] \in \mathbb{R}^{6\times d}$ denote the tuple of all scale and shift parameters, $\mathbf{c} \in \mathbb{R}^d$ and $\mathbf{t} \in \mathbb{R}^d$ represent the embedding for class and time step respectively. For the i-th FiTv2 block, we compute these scale and shift parameters as:

$$\begin{aligned} S^i &= \text{AdaLN}_{\text{global}}(\mathbf{c}+\mathbf{t}) + \text{AdaLN}_{\text{LoRA}}(\mathbf{c}+\mathbf{t}) \\ &= W^{\text{g}}(\mathbf{c}+\mathbf{t}) + W_2^i W_1^i(\mathbf{c}+\mathbf{t}), \end{aligned} \tag{25}$$

where $W^{\text{g}} \in \mathbb{R}^{(6\times d)\times d}, W_2^i \in \mathbb{R}^{(6\times d)\times r}, W_1^i \in \mathbb{R}^{r\times d}$, and the bias parameters are omitted for simplicity. We can adjust the LoRA dimension r to change the parameter ratio in the FiTv2 block. This flexibility allows us to reduce r while simultaneously increasing the number of attention layers N, leading to enhanced model performance. In practice, we set $r = \frac{1}{4}d$ and the final parameters of a FiTv2 block are composed as:

$$\begin{aligned} N &= N_{\text{attn}} + N_{\text{swiglu}} + N_{\text{AdaLN-LoRA}} \\ &= 4 \cdot d^2 + 8 \cdot d^2 + 1.75 \cdot d^2. \end{aligned} \tag{26}$$

Compared with Eq. (24), we decrease the model parameters occupied by the AdaLN module, enabling us to increase N accordingly while maintaining the model parameters in line with the baseline, as shown in Tab. 1.

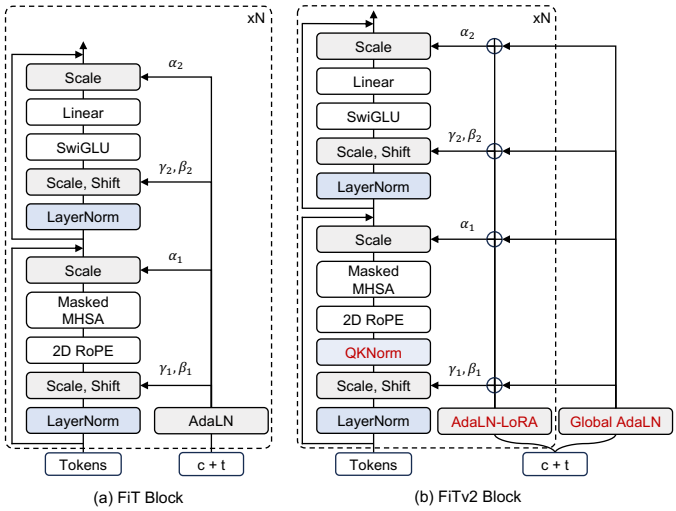


Fig. 5: **Block comparison between (a) FiT and (b) FiTv2.** New modules, QKNorm, AdaLN-LoRA and Global AdaLN, are marked by red color.

4.3.2 Training Strategy

Switching from DDPM to rectified flow. DDPM [14] is a widely used framework for diffusion models, however, it often exhibits limitations in sampling efficiency. Recently, the rectified flow [15] framework proposes a more flexible manner than DDPM which constructs a transport between two distributions through ordinary differential equations. Unlike DDPM relying on discretized time steps, rectified flow follows straight paths, enabling faster simulation. This elimination of time discretization not only enhances sampling efficiency but also simplifies the overall process. Such inherent advantages have enabled the development of advanced generative models, such as SiT [12] and SD3 [54]. We follow the rectified flow implementation in SiT, linearly connecting the noise and data distributions, and predicting the velocity fields.

Mixed data preprocessing. Although the original FiT achieves state-of-the-art performance across unrestricted resolutions and aspect ratios, it underperforms on the standard *ImageNet*-256 $\times$ 256 benchmark. We posit that this discrepancy arises from the methodological difference in dataset preparation, specifically, our initial reliance on image resizing alone, as opposed to the standard resizing and cropping operations employed in the ADM [11] ImageNet reference dataset used for our FID [55] evaluation.

To bridge this gap, we incorporate the fixed-resolution images into our data preprocessing, as shown in Fig. 2 (c). Furthermore, as outlined in Sec. 4.1, to mitigate the blurriness from upscaling low-resolution images, we only crop

Algorithm 1: Mixed Data Preprocessing (LCD)

Input : image $I \in \mathbb{R}^{C,H,W}$, target resolution size S.
if H > S and W > S:
  if random.random() > 0.5:
    return CenterCrop(Resize(I))
  else:
    return Resize(I)
else:
  return Resize(I)

images whose width and height are both larger than the target resolution size. For *ImageNet*-256 × 256 benchmark, only images whose width and height are both larger than 256 may be chosen to be cropped. Exactly, in preprocessing, for images whose sizes are both larger than the target resolution size, we randomly select between resizing only or resizing and cropping with a probability of $\frac{1}{2}$, as in Algorithm 1. For images that do not meet these criteria, we simply resize them to satisfy the sequence length limitation.

As we incorporate resized and cropped images as a subset of our training dataset, we can align the generation distribution of our model with the distribution of the ADM ImageNet reference dataset used for our FID evaluation. Furthermore, the strict restrictions on applying cropping operation and the mixing of flexible images help our model avoid the blurring and information loss problems in previous methods. As a result, this modification enables our FiTv2 to achieve competitive performance on the standard *ImageNet*-256 × 256 benchmark and *ImageNet*-512 × 512 benchmark, while still maintaining the ability to generate images across arbitrary resolutions and aspect ratios.

Improved sampling strategy. Typically, the rectified flow scheduler samples timesteps uniformly from the $[0, 1]$ interval. Recent studies conducted by SD3 [54] have investigated the choice of timestep sampling strategies and found that the Logit-Normal sampler outperforms the original uniform sampler as well as other variants. Formally, the Logit-Normal sampler is defined as:

Statistically, this transformation via the logit function ensures that the tails of normal distribution map to the extremes of the $[0, 1]$ interval in a way that naturally gives more weight to the central part of the diffusion process. Therefore, the logit-normal sampler facilitates the challenge of learning velocity in the middle of the schedule, as highlighted by EDM [56], and significantly accelerates the model convergence.

4.3.3 High-resolution Post-training

Previous state-of-the-art methods typically train high-resolution models from scratch, thus incurring substantial computational costs. We hypothesize that models trained on low-resolution images have already learned the essential semantic information from the *ImageNet* dataset, but have not been adapted to high-resolution. Therefore, we freeze the majority of parameters of the model and adapt this model through additional parameter-efficient fine-tuning on the high-resolution data.

Inspired by BitFit [57], our post-training keeps most parameters of the model frozen, only unfreezing specific parameters related to bias and normalization. Considering the increased image resolution, we also unfreeze the parameters of the image patch embedder and the final output layer, leading to only 14.15% of the overall parameters to be trained. Additionally, we apply the NTK Interpolation to the 2-D RoPE embedding to facilitate the transition to higher resolutions.

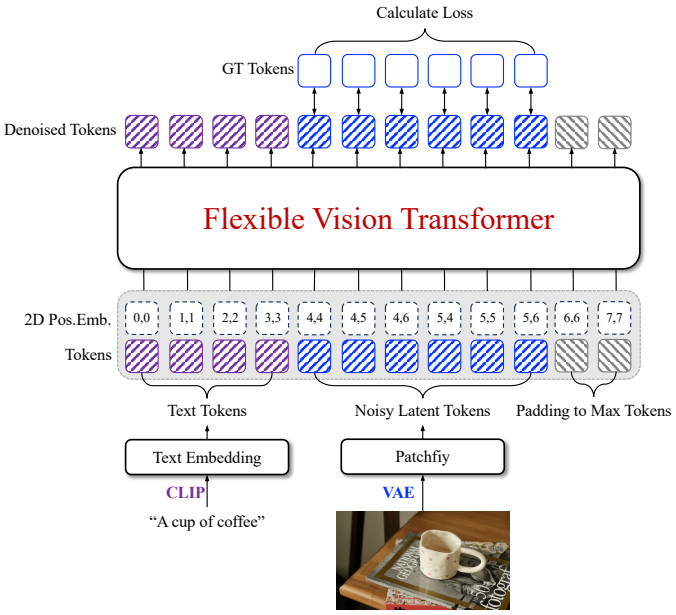


Fig. 6: **Overview of our text-to-image generation model flexible training pipeline.** We utilize CLIP-L to encode text prompts and SD-XL VAE to encode image latents.

4.3.4 Text-to-Image Generation

We further evaluate the effectiveness of our FiTv2 model on the text-to-image [54], [58], [59] (T2I) generation task. As illustrated in Fig. 6, we encode an image into image latents with a pre-trained SDXL-VAE [59] encoder and patchify the image latents to latent tokens. We use CLIP-L [60] text encoder to encode the image caption into text features and embed them into text tokens with an MLP. The FiTv2-T2I model processes the concatenated text tokens and noised latent image latents to predict the denoised latent image tokens. The output text tokens and padding tokens are discarded when calculating loss. To accommodate the 1D text tokens with our 2-D RoPE, we convert each single text positional index to a 2D index tuple. Formally, given text tokens $\mathcal{T} \in \mathbb{R}^{M \times D}$ and latent image tokens $\mathcal{I} \in \mathbb{R}^{(H \times W) \times D}$, the text positional indices and image positional indices are defined as follows:

$$P_{\mathcal{T}} = [(0,0),(1,1)\cdots,(M-1,M-1)],$$
$$P_{\mathcal{I}} = \begin{bmatrix} (M,M) & \cdots & (M,M+W-1) \\ \vdots & \ddots & \vdots \\ (M+H-1,M) & \cdots & (M+H-1,M+W-1) \end{bmatrix}. \tag{27}$$

We also leverage the modulation mechanism of AdaLN module for text conditioning. Specifically, we average-pool the text tokens $\mathcal{T}$ into a semantic text embedding $c_{\mathcal{T}} \in \mathbb{R}^D$,

replacing the original class embedding. This pooled text embedding, along with the time embedding, is then used as input for the global AdaLN and the AdaLN-LoRA modules.

4.4 Training Free Resolution Extrapolation

4.4.1 Vision Positional Interpolation

We denote the inference resolution as (H_{test}, W_{test}). Our FiT can handle various resolutions and aspect ratios during training, so we denote training resolution as $L_{\text{train}} = \sqrt{L_{\max}}$.

By changing the scale factor in Eq. (15) to $s = \max(\max(H_{\text{test}}, W_{\text{test}})/L_{train}, 1.0)$, we can directly implement the positional interpolation methods in large language model extrapolation on 2-D RoPE, which we call vanilla NTK and YaRN implementation. Furthermore, we propose vision RoPE interpolation methods by using the decoupling attribute in decoupled 2-D RoPE. We modify Eq. (20) to:

$$\begin{aligned}\hat{f}_q(\mathbf{q}_m, h_m, w_m) &= [e^{ih_m\Theta_h}\mathbf{q}_m \parallel e^{iw_m\Theta_w}\mathbf{q}_m],\\ \hat{f}_k(\mathbf{k}_n, h_n, w_n) &= [e^{ih_n\Theta_h}\mathbf{k}_n \parallel e^{iw_n\Theta_w}\mathbf{k}_n],\end{aligned} \tag{28}$$

where $\Theta_h = \{\theta_d^h = b_h^{-2d/|D|}, 1 \leqslant d \leqslant \frac{|D|}{2}\}$ and $\Theta_w = \{\theta_d^w = b_w^{-2d/|D|}, 1 \leqslant d \leqslant \frac{|D|}{2}\}$ are calculated separately. Accordingly, the scale factor of height and width is defined separately as

$$s_h = \max(\frac{H_{\text{test}}}{L_{\text{train}}}, 1.0), \quad s_w = \max(\frac{W_{\text{test}}}{L_{\text{train}}}, 1.0). \tag{29}$$

Definition 4.1. *The Definition of VisionNTK Interpolation is a modification of NTK-aware Interpolation by using Eq. (28) with the following rotary base.*

$$b_h = b \cdot s_h^{\frac{|D|}{|D|-2}}, \quad b_w = b \cdot s_w^{\frac{|D|}{|D|-2}}, \tag{30}$$

where $b = 10000$ is the same with Eq. (11)

Definition 4.2. *The Definition of VisionYaRN Interpolation is a modification of YaRN Interpolation by using Eq. (28) with the following rotary frequency.*

$$\begin{aligned}\theta_d^h &= (1 - \gamma(r(d))\frac{\theta_d}{s_h} + \gamma(r(d))\theta_d,\\ \theta_d^w &= (1 - \gamma(r(d))\frac{\theta_d}{s_w} + \gamma(r(d))\theta_d,\end{aligned} \tag{31}$$

where $\gamma(r(d))$ is the same with Eq. (16).

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, see Sec. 5.4.

4.4.2 Attention Scale for Longer Context

In the context of resolution-extrapolation, another approach beyond positional embedding interpolation is scaling the attention logits to aggregate information effectively. Previous studies [50], [51] have theoretically demonstrated that longer contexts result in higher attention entropy of models trained on shorter contexts, leading to widespread aggregation for each token. For higher-resolution image generation, this can cause redundancy in spatial information and disordered object presentations, thereby destroying aesthetics and fidelity. Therefore, a scale factor, which is defined as $s = max(1.0, \sqrt{\log \frac{H_{\text{test}} \times W_{\text{test}}}{H_{\text{train}} \times W_{\text{train}}}})$, is introduced to mitigate the entorpy fluctuations. The formulation of scaled attention is as follows:

$$\text{Softmax}(\frac{1}{\sqrt{d_k}}\text{LN}(Q_i)\text{LN}(K_i)^T \cdot s + M). \tag{32}$$

Model	Layers N	Hidden size d	Heads	Params	GFLOPs
SiT-B	12	768	12	131M	21.8
FiT-B	12	768	12	159M	29.1
FiTv2-B	15	768	12	128M	27.3
SiT-XL	28	1152	16	675M	114
FiT-XL	28	1152	16	824M	153
FiTv2-XL	36	1152	16	671M	147
FiTv2-3B	40	2304	24	3B	653

TABLE 1: **Details of FiTv2 model architecture.** We follow our original FiT to set the base model and XL model for FiTv2. We also scale up our FiTv2 to 3 billion parameters as our largets model.

5 Experiments

5.1 FiT Implementation

We present the implementation details of FiTv2, including model architecture, training details, and evaluation metrics.
Model architecture. The detailed model architecture is shown in Tab. 1. For FiT, we follow SiT-B and SiT-XL to set the same layers, hidden size, and attention heads for base model FiT-B and x-large model FiT-XL. For FiTv2, as described in Sec. 4.3.1, we reassign parameters to increase the model layers, thereby aligning the parameters with those of DiT [2] and SiT [12]. As SiT reveals stronger synthesis performance when using a smaller patch size, we use a patch size p=2, denoted by FiT-B/2 and FiTv2-B/2. We adopt the same off-the-shelf pre-trained VAE [10] as SiT, which is provided by the Stable Diffusion [10] to encode/decode the image/latent tokens. The VAE encoder has a downsampling ratio of $1/8$ and a feature channel dimension of 4. An image of size $160 \times 320 \times 3$ is encoded into latent codes of size $20 \times 40 \times 4$. The latent codes of size $20 \times 40 \times 4$ are patchified into latent tokens of length $L = 10 \times 20 = 200$.
Training details. We train class-conditional latent FiTv2 models under predetermined maximum resolution limitation, i.e., $H \cdot W \leqslant 256^2$ (equivalent to token length $L \leqslant 256$) for pre-training and $H \cdot W \leqslant 512^2$ (equivalent to token length $L \leqslant 1024$) for post-training, on the *ImageNet* [1] dataset. We down-resize the high-resolution images to meet the $HW \leqslant 256^2$ limitation while maintaining the aspect ratio. We follow SiT to use Horizontal Flip Augmentation. For the pre-training process, we employ a linear learning rate warm-up over the first 5000 steps for stability. Subsequently, we use a constant learning rate of 1×10^{-4} using AdamW [61], no weight decay, and a batch size of 256, consistent with SiT. To reduce the training costs, all the experiments are conducted using mixed-precision training.

Following common practice in the generative modeling literature, we adopt an exponential moving average (EMA)

Arch.	Pos. Embed.	FFN	Train	256×256 (i.d.) FID↓	sFID↓	IS↑	Prec.↑	Rec.↑	160×320 (i.d.) FID↓	sFID↓	IS↑	Prec.↑	Rec.↑	224×448 (o.o.d.) FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
DiT-B/2	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/2	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

TABLE 2: **Ablation results from DiT-B/2 to FiT-B/2 without using classifier-free guidance.** We train all the models to $400k$ steps for fair comparision.

Method	Scheduler	QK-Norm	Parameters	Data	Sampling	256×256 (400k) cfg=1.0	cfg=1.5	256×256 (1000k) cfg=1.0	cfg=1.5	256×256 (1500k) cfg=1.0	cfg=1.5	256×256 (2000k) cfg=1.0	cfg=1.5
DiT-B/2	DDPM	-	-	-	-	45.33	22.21	33.27	12.59	✗	✗	✗	✗
SiT-B/2	Rectified Flow	-	-	-	-	36.7	16.31	27.13	9.3	✗	✗	✗	✗
FiT-B/2	DDPM	No	Original	Flexible	Uniform	36.36	18.86	29.14	11.06	26.08	9.23	✗	✗
Config E	Rectified Flow	No	Original	Flexible	Uniform	30.74	13.14	23.48	8.67	22.32	8.25	21.23	7.61
Config F	Rectified Flow	LayerNorm	Original	Flexible	Uniform	30.83	13.21	23.64	8.57	21.64	7.70	20.73	7.10
Config G	Rectified Flow	LayerNorm	Reassigned	Flexible	Uniform	28.59	12.74	21.16	8.05	19.56	7.16	18.42	6.60
Config H	Rectified Flow	No	Original	Mixed	Uniform	34.15	13.99	25.54	8.27	23.63	7.24	✗	✗
Config I	Rectified Flow	LayerNorm	Original	Mixed	Uniform	34.55	14.19	25.94	8.37	23.45	6.99	22.04	6.31
Config J	Rectified Flow	LayerNorm	Original	Mixed	Logit-Normal	28.49	9.98	21.93	6.16	20.09	5.23	19.21	4.84
FiTv2-B/2	Rectified Flow	LayerNorm	Reassigned	Mixed	Logit-Normal	**26.03**	**9.45**	**19.02**	**5.51**	**17.70**	**4.73**	**16.52**	**4.30**

TABLE 3: **Ablation results from FiT-B/2 to FiTv2-B/2 without using classifier-free guidance.** We train the models to $2000k$ steps to assess stability. A ✗ indicates that the training process breaks down before reaching this evaluation point.

Method	320×320 (1:1) FID↓	sFID↓	IS↑	Prec.↑	Rec.↑	224×448 (1:2) FID↓	sFID↓	IS↑	Prec.↑	Rec.↑	160×480 (1:3) FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
SiT-XL/2	19.72	54.91	144.06	0.63	0.47	46.17	67.89	73.32	0.43	0.43	104.57	91.47	23.43	0.16	0.41
SiT-XL/2 + EI	8.93	19.68	212.99	0.72	0.5	78.87	48.97	43.57	0.27	0.45	131.04	71.18	17.63	0.11	0.43
SiT-XL/2 + PI	8.55	20.74	217.74	0.73	0.49	82.51	50.83	41.67	0.26	0.44	133.47	72.81	17.57	0.11	0.43
FiTv2-XL/2	5.79	13.7	233.03	0.75	0.55	10.46	17.24	184.06	0.68	0.54	16.4	19.55	127.72	0.59	0.51
FiTv2-XL/2 + PI	11.47	21.131	197.04	0.67	0.51	154.59	77.21	13.18	0.10	0.14	169.4	9.81	78.31	0.06	0.06
FiTv2-XL/2 + YaRN	5.87	15.38	250.66	0.77	0.52	21.41	34.70	146.31	0.56	0.38	36.73	35.81	78.55	0.42	0.26
FiTv2-XL/2 + NTK	6.04	14.35	232.91	0.75	0.55	10.82	17.84	184.68	0.66	0.53	16.3	20.13	131.8	0.58	0.50
FiTv2-XL/2 + VisionYaRN	5.87	15.38	250.66	0.77	0.52	6.62	18.22	245.47	0.76	0.48	16.17	27.35	**151.99**	0.62	0.39
FiTv2-XL/2 + VisionNTK	6.04	14.35	232.91	0.75	**0.55**	10.11	17.08	188.4	0.68	**0.53**	15.44	19.48	135.57	0.60	0.50
FiTv2-XL/2 + VisionNTK + Attn-Scale	**3.55**	**9.60**	**274.48**	**0.82**	0.52	**5.54**	**14.53**	**233.11**	**0.77**	0.51	**13.55**	**19.47**	144.62	**0.63**	**0.50**

TABLE 4: **Benchmarking class-conditional image generation with out-of-distribution resolution on ImageNet.** The official SiT-XL/2 at $7000k$ training steps and our FiTv2-XL/2 at $2000k$ training steps are adopted in this experiment. Metrics are calculated using classifier-free guidance (cfg=1.5). YaRN and NTK mean the vanilla implementation of such two methods. Our FiTv2-XL/2 demonstrates stable extrapolation performance, which can be significantly improved combined with VisionNTK and attention scale methods.

of model weights over training with a decay of 0.9999. All results are reported using the EMA model. We retain the same rectified flow hyper-parameters as SiT.

Evaluation details and metrics. We evaluate models with some commonly used metrics, *i.e.* Fre'chet Inception Distance (FID) [62], sFID [55], Inception Score (IS) [63], improved Precision and Recall [64]. For fair comparisons, we follow DiT to use the TensorFlow evaluation from ADM [11] to report FID-50K and other results. FiT and DiT are sampled with 250 DDPM sampling steps, while FiTv2 and SiT both use the adaptive-step ODE sampler (i.e., dopri5) to generate images. FID is used as the major metric as it measures both diversity and fidelity. We additionally report IS, sFID, Precision, and Recall as secondary metrics. For the FiT architecture experiment (Sec. 5.2 and Tab. 2) we report the results without using classifier-free guidance [65]. For other experiments, we report the exact CFG scale if used. The ablation evaluation results on the CFG scale of our FiTv2 model are shown in Fig. 8.

Evaluation resolution. Unlike previous work that mainly conducted experiments on a fixed aspect ratio of $1:1$, we conducted experiments on different aspect ratios, which are $1:1$, $1:2$, and $1:3$, respectively. On the other hand, we divide the experiment into resolution within the training distribution and resolution out of the training distribution. For the resolution in distribution, we mainly use 256×256 (1:1), 160×320 (1:2), and 128×384 (1:3) for evaluation, with 256, 200, 192 latent tokens respectively. All token lengths are smaller than or equal to 256, leading to respective resolutions within the pre-training distribution. For the resolution out of distribution, we mainly use 320×320 (1:1), 224×448 (1:2), and 160×480 (1:3) for evaluation, with 400, 392, 300 latent tokens respectively. All token lengths are larger than 256, resulting in the resolutions out of pre-training distribution. Through such division, we holistically evaluate the image synthesis and resolution extrapolation ability of FiTv2 at various resolutions and aspect ratios.

5.2 FiT Architecture Design

In this part, we conduct an ablation study to verify the architecture designs in FiT. We report the results of various variant FiT-B/2 models at 400K training steps and use FID-50K, sFID, IS, Precision, and Recall as the evaluation metrics. We conduct experiments at three different resolutions: 256×256, 160×320, and 224×448. These resolutions are

Method	Images	Params	256×256 (1:1) FID↓	sFID↓	IS↑	Prec.↑	Rec.↑	160×320 (1:2) FID↓	sFID↓	IS↑	Prec.↑	Rec.↑	128×384 (1:3) 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	355M	-	6.18	-	182.1	0.80	0.51	-	-	-	-	-	-	-	-	-	-
CDM	-	-	4.88	-	158.71	-	-	-	-	-	-	-	-	-	-	-	-
Large-DiT-7B	256M	7.3B	6.09	5.59	153.32	0.70	0.59	-	-	-	-	-	-	-	-	-	-
Efficient-DiT-G (cfg=1.5)	-	675M	2.01	*4.49*	271.04	0.82	0.60	-	-	-	-	-	-	-	-	-	-
MaskDiT-G	2048M	-	2.28	5.67	276.56	0.80	**0.61**	-	-	-	-	-	-	-	-	-	-
SimpleDiffusion-G (cfg=1.1)	1024M	2B	2.44	-	256.3	-	-	-	-	-	-	-	-	-	-	-	-
Flag-DiT-3B-G*	256M	4.23B	1.96	**4.43**	284.8	0.82	**0.61**	-	-	-	-	-	-	-	-	-	-
Large-DiT-3B-G*	435M	4.23B	2.10	4.52	**304.36**	0.82	0.60	118.98	62.00	12.24	0.14	0.28	142.76	80.62	10.74	0.075	0.26
U-ViT-H/2-G (cfg=1.4)	512M	501M	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	507M	673M	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)	214M	395M	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)	1664M	676M	**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=1.5)	1792M	675M	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
SiT-XL/2-G (cfg=1.5)	1792M	675M	2.15	4.50	258.09	0.81	0.60	17.38	28.59	110.32	0.52	0.65	87.40	57.41	23.45	0.16	0.56
FiT-XL/2-G (cfg=1.5)	512M	824M	4.21	10.01	254.87	0.84	0.51	**5.48**	**9.95**	192.93	0.74	0.56	16.59	**20.81**	111.59	0.57	0.52
FiTv2-XL/2-G (cfg=1.5)	512M	671M	2.26	4.53	260.95	0.81	0.59	5.50	11.42	211.26	0.74	0.55	14.46	23.20	135.31	0.60	0.47
FiTv2-3B/2-G (cfg=1.5)	256M	3B	2.15	4.49	276.32	0.82	0.59	6.72	13.13	**233.31**	**0.76**	0.50	**13.73**	**23.26**	**145.38**	**0.61**	0.48

TABLE 5: **Benchmarking class-conditional image generation with in-distribution resolution on *ImageNet* dataset.** "-G" denotes the results with classifier-free guidance. *: Flag-DiT-3B and Large-DiT-3B actually have 4.23 billion parameters, where 3B means the parameters of all transformer blocks. †: MDT-G adpots an improved classifier-free guidance strategy: $w_t = (1 - \cos \pi(\frac{t}{t_{max}})^s)w/2$, where $w = 3.8$ is the maximum guidance scale and $s = 4$ is the controlling factor.

Method	Images	Params	320×320 (1:1) FID↓	sFID↓	IS↑	Prec.↑	Rec.↑	224×448 (1:2) FID↓	sFID↓	IS↑	Prec.↑	Rec.↑	160×480 (1:3) FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
U-ViT-H/2-G (cfg=1.4)	512M	501M	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	507M	774M	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)	214M	395M	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
DiT-XL/2-G (cfg=1.5)	1792M	675M	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	0.09	0.45
SiT-XL/2-G (cfg=1.5)	1792M	675M	8.55	20.74	217.74	0.73	0.49	82.51	50.83	41.67	0.26	0.44	133.5	72.81	17.57	0.11	0.43
FiT-XL/2-G (cfg=1.5)	512M	824M	5.11	13.32	256.15	0.81	0.47	7.60	17.15	218.74	0.74	0.47	15.20	20.96	135.17	0.62	0.48
FiTv2-XL/2-G* (cfg=1.5)	512M	671M	3.55	9.60	274.48	0.82	**0.55**	5.54	14.53	233.11	0.77	**0.51**	13.55	**19.47**	144.62	0.63	0.50
FiTv2-3B/2-G* (cfg=1.5)	256M	3B	**3.22**	9.96	**291.13**	**0.83**	0.53	**4.87**	**14.47**	**263.27**	**0.80**	0.49	**12.15**	**19.47**	**162.24**	**0.65**	0.48

TABLE 6: **Benchmarking class-conditional image generation with out-of-distribution resolution on *ImageNet* dataset.** *: FiTv2 adopts VisionNTK and attention scale for resolution extrapolation. Our FiTv2 model achieves state-of-the-art performance across all the resolutions and aspect ratios, demonstrating a strong extrapolation capability.

Method	Images	Params	512×512 (1:1) FID↓	sFID↓	IS↑	Prec.↑	Rec.↑	320×640 (1:2) FID↓	sFID↓	IS↑	Prec.↑	Rec.↑	256×768 (1:3) FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
DiM-Huge-G (cfg=1.7)	+26M	860M	3.78	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DiffusionSSM-XL-G	302M	660M	3.41	5.84	255.06	0.85	0.49	-	-	-	-	-	-	-	-	-	-
MaskGiT	384M	227M	7.32	-	156.0	0.78	0.50	-	-	-	-	-	-	-	-	-	-
SimpleDiffusion-G (cfg=1.1)	1024M	2B	3.02	-	248.7	-	-	-	-	-	-	-	-	-	-	-	-
DiffiT-G (cfg=1.49)	-	561M	2.67	-	252.12	0.83	0.55	-	-	-	-	-	-	-	-	-	-
MaskDiT-G	1024M	-	2.50	5.10	256.27	0.83	0.56	-	-	-	-	-	-	-	-	-	-
Large-DiT-3B-G (cfg=1.5)	471M	4.23B	2.52	5.01	**303.70**	0.82	0.57	-	-	-	-	-	-	-	-	-	-
U-ViT-H/2-G (cfg=1.4)	512M	501M	4.05	6.44	263.79	**0.84**	0.48	9.79	14.64	188.8	0.76	0.49	146.58	78.69	12.47	0.21	0.36
ADM-G,U	1385M	774M	3.85	5.86	221.72	**0.84**	0.53	13.31	**10.67**	113.69	0.73	**0.64**	33.35	25.04	59.23	0.61	**0.62**
DiT-XL/2-G (cfg=1.5)	768M	675M	3.04	**5.02**	240.82	**0.84**	0.54	41.25	66.83	54.84	0.54	0.59	148.25	154.39	6.64	0.13	0.36
FiTv2-XL/2-G (cfg=1.65)	+102M	671M	2.90	5.73	263.11	0.83	0.53	4.87	10.75	228.09	**0.80**	0.53	18.55	21.69	126.55	0.69	0.53
FiTv2-3B/2-G (cfg=1.6)	+51M	3B	**2.41**	5.34	284.49	0.82	**0.58**	**4.54**	11.04	**240.30**	**0.80**	0.56	**16.08**	**19.75**	**140.10**	**0.72**	0.52

TABLE 7: **Benchmarking class-conditional image generation with high-resolution image generation on *ImageNet* dataset.** Our FiTv2 can directly generates images with different aspect ratios with stable and state-of-the-art performance.

chosen to encompass different aspect ratios, as well as to include resolutions both in and out of the distribution.

Flexible training vs. Fixed training. *Flexible training pipeline significantly improves the performance across various resolutions.* This improvement is evident not only within the in-distribution resolutions but also extends to resolutions out of the training distribution, as shown in Tab. 2. *Config A* is the original DiT-B/2 model only with flexible training, which slightly improves the performance (**-1.49** FID) compared with DiT-B/2 with fixed resolution training at 256×256 resolution. *Config A* demonstrates a significant performance improvement through flexible training. Compared to DiT-B/2, FID scores are reduced by **40.81** and **56.55** at resolutions 160×320 and 224×448, respectively.

SwiGLU vs. MLP. *SwiGLU slightly improves the performance across various resolutions, compared to MLP. Config B* is the FiT-B/2 flexible training model replacing MLP with SwiGLU. Compared to *Config A*, *Config B* demonstrates notable improvements across various resolutions. Specifically, for resolutions of 256×256, 160×320, and 224×448, *Config B* reduces the FID scores by **1.59**, **1.85**, and **0.21** in Tab. 2, respectively. So FiT uses SwiGLU in FFN.

2D RoPE vs. Absolute PE. *2D RoPE demonstrates greater efficiency compared to absolute position encoding, and it possesses significant extrapolation capability across various resolutions. Config D* is the FiT-B/2 flexible training model replacing absolute PE with 2D RoPE. For resolutions within the training distribution, specifically 256×256 and 160×320, *Config D* reduces the FID scores by **6.05**, and **5.45** in Tab. 2, compared to *Config A*. For resolution beyond the training distribution, 224×448, *Config D* shows significant extrapolation capability (**-6.39** FID) compared to *Config A*. *Config C* retains both absolute PE and 2D RoPE. However, in a comparison between *Config C* and *Config D*, we observe that *Config C* performs worse. For resolutions of 256x256, 160x320, and 224x448, *Config C* increases FID scores of **1.82**, **1.65**, and **0.44**,

respectively, compared to *Config D*. Therefore, only 2D RoPE is used for positional embedding in our implementation.

Putting it together. *FiT demonstrates significant and comprehensive superiority across various resolution settings, compared to original DiT.* FiT has achieved state-of-the-art performance across various configurations. Compared to DiT-B/2, FiT-B/2 reduces the FID score by **8.47** on the most common resolution of 256×256 in Tab. 2. Furthermore, FiT-B/2 has made significant performance gains at resolutions of 160×320 and 224×448, decreasing the FID scores by **47.36** and **64.43**, respectively.

5.3 From FiT to FiTv2

In this section, we conduct an ablation study to validate the architecture design in FiTv2. We report the results of various variants of FiTv2-B/2, utilizing FID at 256×256 resolution, and compare these with the DiT-B/2, and SiT-B/2. We train all the models to $2000K$ steps to access the training stability.

Rectified Flow vs. DDPM. *Rectified Flow scheduler significantly improves the performance and training stability in our FiT model.* Specifically, *Config E* replaces the DDPM scheduler in the original FiT-B/2 with the rectified flow scheduler, leading to substantial performance improvement, both with and without classifier-free guidance (CFG), as in Tab. 3. Notably, *Config E* successfully trains to $2000K$ steps, while the training of FiT-B/2 fails after $1500K$ steps, highlighting the stability benefits of the rectified flow scheduler.

QK-Norm vs. No Norm. *QK-Norm contributes to stabilizing the training process and provides a slight performance enhancement.* We implement LayerNorm for the query and key vectors of attention (*Config F* and *Config I* compared to *Config E* and *Config H*, respectively). As in Tab. 3, *Config F* generally achieves better FID scores than *Config E*. Remarkably, we observe that *Config I* maintains a stable training process up to $2000K$ steps, while *Config H* fails to reach this training step. Furthermore, *Config I* outperforms *Config H* at $1500K$ steps in terms of FID score.

Reassigned parameters vs Original parameters. *Parameter reassignment enhances the efficiency and effectiveness of our FiTv2 model.* As detailed in Sec. 4.3.1, we reassign the parameters in FiTv2 to optimize the architecture, comparing the reassigned parameters (FiTv2-B/2) with the original parameters (FiT-B/2) in Tab. 3. *Config G*, which adopts the reassigned parameters, shows consistent FID improvements across all evaluation points compared with *Config F*.

Mixed training vs. Flexible training. *Mixed training improves the model performance when using CFG.* As shown in Tab. 3, *Config I* employs a mixed training strategy and exhibits FID performance gains at 1000k, 1500k, and 2000k steps compared to *Config F*.

Logit-Normal sampling vs. Uniform sampling. *Logit-Normal sampling significantly accelerates the convergence speed, compared with uniform sampling.* As demonstrated in Tab. 3, *Config J* obtains better results than *Config I* at all evaluation points, both with and without CFG.

From FiT to FiTv2. *FiTv2 demonstrates significant superiority compared with the original FiT, as well as DiT and SiT.* As reported in Tab. 3, experiments on DiT and SiT both break down after $1000K$ training steps, revealing the instability of their architecture. In contrast, FiTv2 exhibits superior training stability, as well as achieves an approximately $2\times$ faster convergence speed compared with FiT, DiT, and SiT.

5.4 Resolution Extrapolation Design

In this part, we adopt the official SiT-XL/2 model at $7000K$ training steps and our FiTv2-XL/2 model at $2000K$ training steps to evaluate the extrapolation performance on three out-of-distribution resolutions: 320×320, 224×448 and 160×480. Direct extrapolation does not perform well on larger resolutions outside of training distribution. So we conduct a comprehensive benchmarking analysis focused on higher resolution extrapolation.

PI and EI. PI (Position Interpolation) and EI (Embedding Interpolation) are two baseline positional embedding interpolation methods for resolution extrapolation. PI linearly down-scales the inference position coordinates to match the original coordinates. EI resizes the positional embedding with bilinear interpolation. Following ViT [30], EI is used for absolute positional embedding.

NTK and YaRN. We set the scale factor to $s = \max(\max(H_{\text{test}}, W_{\text{test}})/\sqrt{256})$ and adopt the vanilla implementation of the two methods, as in Sec. 3.2. For YaRN, we set $\alpha = 1, \beta = 32$ in Eq. (17).

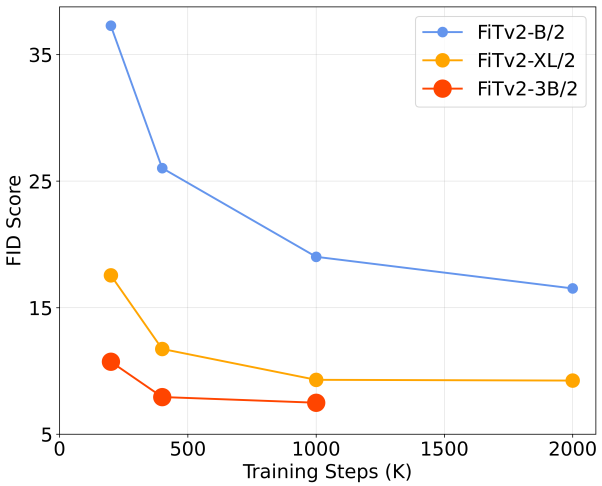
VisionNTK and VisionYaRN. These two methods are defined detailedly in Theorems 4.1 and 4.2. Note that when the aspect ratio equals one, VisionNTK and VisionYaRN are equivalent to NTK and YaRN, respectively.

Attention Scale. The attention scale is defined in Sec. 4.4.2, we apply this technique combined with the VisonNTK.

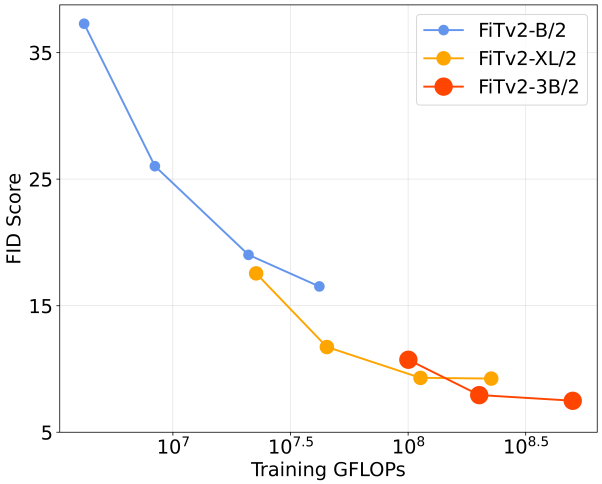
Analysis. We present in Tab. 4 that our FiTv2-XL/2 shows stable performance when directly extrapolating to larger resolutions. When combined with PI, the extrapolation performance of FiTv2-XL/2 at all three resolutions decreases. When directly combined with YaRN, the FID score on 320×320 changes slightly, but the performance on 224×448 and 168×480 descends. Our VisionYaRN solves this dilemma and reduces the FID score by **3.84** on 224×448 compared with YaRN. NTK interpolation method demonstrates stable extrapolation performance but increases the FID score slightly at 320×320 and 224×448 resolutions. Our VisionNTK method slightly exceeds the performance of direct extrapolation on 224×448 and 160×480 resolutions. When combining VisionNTK and attention scale, the performance significantly surpasses all the other extrapolation methods, with FID improvement **2.24** on 320×320, **4.92** on 224×448 and **2.89** on 160×480 compared with direct extrapolation.

In conclusion, our FiTv2-XL/2 model demonstrates robust extrapolation capabilities. Additionally, VisionYaRN and VisonNTK can enhance the generation performance on varied aspect ratios. Furthermore, the combination of VisionNTK with attention scale greatly improves high-resolution extrapolation ability.

However, the official SiT-XL/2 model demonstrates poor extrapolation ability, in Tab. 4. When combined with PI, the FID score achieves **19.72** at 320×320 resolution, which still falls behind our FiTv2-XL/2. At 224×448 and 160×480 resolutions, PI and EI interpolation methods cannot improve the extrapolation performance.

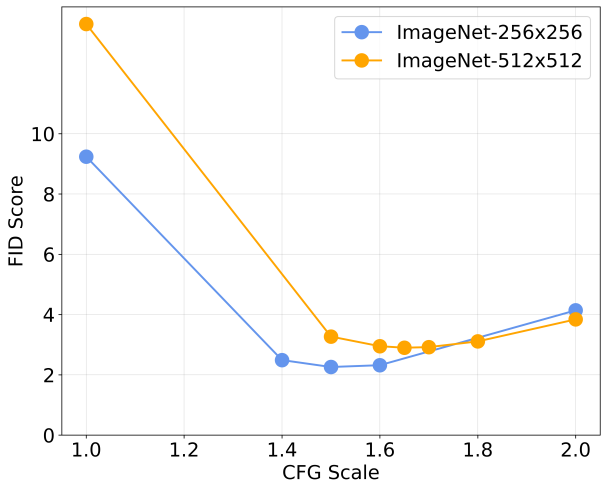


(a) FID Score vs. Training Steps.

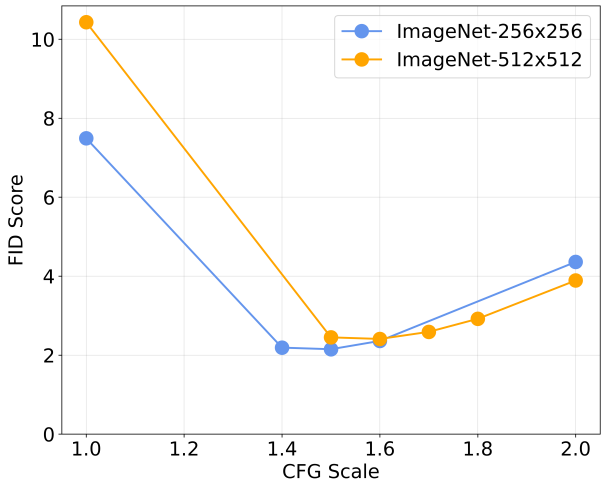


(b) FID Score vs. Training Steps.

Fig. 7: **Effect of scaling FiTv2 model.** All the images are sampled without using CFG. We demonstrate FID over training iterations (a) and training GFLOPs (b) of our FiTv2 model of three sizes. Scaling our FiTv2 model yieds better quantitative and qualitative performance.



(a) FID Score vs. CFG Scale of FiTv2-XL/2.



(b) FID Score vs. CFG Scale of FiTv2-3B/2.

Fig. 8: **Effect of classifier-free guidance scale on FID score for *ImageNet*-256 $\times$ 256 and *ImageNet*-512 $\times$ 512 experiments with (a) FiTv2-XL/2 and (b) FiTv2-3B/2 models.** (a) For FiTv2-XL/2 model, the optimal performance is achieved with CFG=1.5 for 256 $\times$ 256 resolution and CFG=1.65 for 512 $\times$ 512 resolution. (b) For FiTv2-3B/2 model, the optimal performance is observed with CFG=1.5 for 256 $\times$ 256 resolution and CFG=1.6 for 512 $\times$ 512 resolution.

5.5 Pre-trained Model Results

5.5.1 *In-Distribution Resolution Results*

In this part, we compare our FiTv2 model with other baselines. Our FiTv2-XL model is trained with $2000K$ steps, consuming only 28.6% of the cost of SiT but with better performance. Furthermore, we scale our FiTv2 up to 3B parameters, which is trained with $1000K$ steps. We conduct experiments to evaluate the performance of FiTv2 at three different in-distribution resolutions: 256×256, 160×320, and 128×384. We show samples from the FiTv2 in Fig 1, and we compare against some state-of-the-art class-conditional generative models: BigGAN [66], StyleGAN-XL [67], MaskGIT [68], CDM [69], Large-DiT [41], MaskDiT [70], Efficient-DiT [71], SimpleDiffusion [72], Flag-DiT [41], U-ViT [39], ADM [11], LDM [10], MDT [38], DiT [2], SiT [12] and our original FiT. When generating images of 160×320 and 128×384 resolution, we adopt PI on the positional embedding of the DiT and SiT model, as stated in Sec. 5.4. EI is employed in the positional embedding of U-ViT and MDT models, as they use learnable positional embedding. ADM and LDM can directly synthesize images with resolutions different from the training resolution. For Large-DiT, we directly generate images of different resolutions as it uses 1D-RoPE as position embedding. For the FiT and FiTv2 models, we directly generate images with different aspect ratios without any extrapolation techniques.

As shown in Tab. 5, FiTv2-XL/2 and FiTv2-3B/2 outperform all prior diffusion models, demonstrating exceptional performance on both the standard 256×256 benchmark and varied resolutions. FiTv2-XL/2 reduces the FID by 1.95 compared to the original FiT-XL/2 with the same training steps and a smaller model size. Our FiTv2-XL/2 and FiTv2-3B/2 can be competitive with any other SOTA methods on

Fig. 9: **Selected samples from FiTv2-XL/2 models at resolutions of** 256×256 **on text-to-image generation tasks.** All the images are sampled with CFG=4.0. With only $400K$ training steps, our model is capable of generating releastic images according to text descriptions.

256×256 resolution. FiT-XL/2 and FiTv2-XL/2 achieve superior performance on 160×320 resolution, decreasing the previous best FID of **6.93** achieved by U-ViT-H/2-G to **5.48** and **5.50** respectively. On 128×384 resolution, FiTv2-XL/2 and FiTv2-3B/2 show significant superiority, decreasing the previous SOTA FID-50K of **29.67** achieved by LDM-4/G to **14.46** and **13.73** respectively. In conclusion, these results suggest that our FiTv2 model has improved performance on standard benchmarks while maintaining the enhanced capability to generate images with arbitrary aspect ratios.

5.5.2 Out-of-Distribution Resolution Results

We evaluate our FiTv2-XL/2 on three different out-of-distribution resolutions: 320×320, 224×448, and 160×480 and compare against some SOTA class-conditional generative models: U-ViT, ADM, LDM-4, MDT, DiT, SiT, and the original FiT. PI is employed in DiT and SiT, while EI is adopted in U-ViT, as in Sec. 5.5.1. U-Net-based methods, such as ADM and LDM-4 can directly generate images with resolution out of distribution. VisionNTK is adopted in FiT, and we combine VisionNTK and attention scale to our FiTv2 model. Note that we do not evaluate the MDT and Large-DiT, as they fall short of generating images whose resolution differs from the training resolution in Tab. 5.

As shown in Tab. 6, FiTv2-XL/2 and FiTv2-3B/2 achieve the best FID-50K, IS, and Precision, on all three resolutions, indicating their outstanding extrapolation ability. In terms of other metrics, such as sFID and Recall, the FiTv2 model demonstrates competitive performance. FiTv2-XL/2 surpasses FiT-XL/2 on all three resolutions with fewer parameters and FLOPs. Compared with the previous SOTA LDM-4, FiTv2-3B/2 gains FID improvement by **3.02**, **3.68** and **7.09** on 320×320, 224×448 and 160×480 resolutions, respectively.

5.5.3 Analysis of the Pretraining Results

Scalability analysis. In Fig. 7a, we demonstrate how model performance changes as training steps increase. In Fig. 7b, we present the relation of model performance with the training GFLOPs, which is calculated as GFLOPs $\times$ batch size $\times$ training steps $\times$ 3, following DiT [2]. As the GFLOPs increase, whether by increasing training steps or enlarging the model size, the FID score and aesthetic quality consistently improve. Additionally, we observe that with the same training GFLOPs, the larger FiTv2 model always shows better qualitative and quantitative results. In contrast, the smaller FiTv2 model, even when trained for more steps, fails to reach the performance of larger FiTv2 models trained for fewer steps. We conclude that scaling model size is a more efficient approach to managing to compute costs, consistent with the findings from DiT.

Flexibility analysis. LDMs with transformer backbones are known to have difficulty in generating images out of training resolution, such as DiT, U-ViT, MDT, SiT, and Large-DiT. More seriously, MDT almost has no ability to generate images beyond the training resolution. We speculate this is because both learnable absolute PE and learnable relative PE are used in MDT. Large-Dit also encounters difficulty in generating images with varied resolutions, as the usage of 1D-RoPE makes it hard to encode spatial structure in images. DiT, U-ViT, and SiT show a certain degree of extrapolation ability and achieve FID scores of **9.98**, **7.65** and **8.55** respectively at 320×320 resolution. However, when the aspect ratio is not equal to one, their generation performance drops significantly, as 128×384, 224×448, and 160×480 resolutions. Benefiting from the advantage of the local receptive field of the Convolution Neural Network, ADM and LDM show stable performance on resolution extrapolation and generalization ability to various aspect

ratios. Our FiTv2 model solves the problem of insufficient extrapolation and generalization capabilities of the transformer in image synthesis. At 160×320, 128×384, 320×320, 224×448, and 160×480 resolutions, FiTv2-XL/2 exceeds the previous SOTA CNN methods, like ADM and LDM.

5.6 High-resolution Post-trained Model Results

We extend the context length to 1024 (equivalent to $H\cdot W\leqslant 512^2$) to conduct high-resolution post-training. As detailed in Sec. 4.3.3, we utilize the model pre-trained with the context length $L\leqslant 256$, keeping the major parameters frozen. We only update the parameters associated with bias, normalization, image patch embedder, and the final layer, leading to merely 14.15% of the overall parameters. Training is conducted using a constant learning rate of 1×10^{-4} using AdamW, no weight decay, and a batch size of 256, same with the DiT and SiT training setting. Specifically, we train the FiTv2-XL/2 model for $200K$ steps and the FiTv2-3B/2 model for $100K$ steps.

The model performance is evaluated on three resolutions: 512×512 (1:1), 320×320 (1:2), and 256×768 (1:3), offering a comprehensive assessment of the image synthesis capability. Our FiTv2 is compared with several state-of-the-art baselines, including DiM [73], Diffusion-SSm [74], MaskGiT [68], SimpleDiffusion [72], DiffiT [40], MaskDiT [70], Large-DiT [41], U-ViT [39], ADM [11], and DiT [2]. The open-source baseline models are evaluated on 320×640 and 256×768 resolutions. Consistent with Sec. 5.5.1, PI is adopted in DiT while EI is employed in U-ViT. For ADM and our FiTv2, images with different resolutions are directly generated.

As demonstrated in Tab. 7, FiTv2-XL/2 beats DiT-XL/2 on all three resolutions, with comparable parameters and significantly lower training costs. Remarkably, our FiTv2-XL/2 surpasses DiT-XL/2 on the FID score by **36.38** at 320×640 resolution and by **129.7** at 256×768 resolution. Furthermore, our FiTv2-3B/2 consistently outperforms all other baseline models on all three resolutions. FiTv2-3B/2 surpasses the previous SOTA Large-DiT-3B and MaskDiT at 512×512 resolution. At 320×640 and 256×768 resolutions, FiTv2-3B/2 demonstrates significant superiority, exceeding the previous SOTA U-ViT by **5.25** at 320×640 resolution on FID score and surpassing previous SOTA ADM by **17.27** at 256×768 resolution.

5.7 Text-to-Image Results

We conduct text-to-image (T2I) generation experiments to further evaluate the effectiveness of our FiTv2 architecture. We use the filtered and recaptioned CC12M [75] subset from PixelProse [76] for training, which comprises 8.6 million high-quality images with descriptive captions. The CLIP-L [60] text encoder is employed to extract text features, resulting in 77 text tokens, each with 768 dimensions. We use the penultimate hidden representation from the CLIP-L text encoder as the text features following Imagen [77]. We use the SDXL-VAE [59] to extract image latents and the training pipeline follows the class-guided image generation methodology described in Sec. 4.1. The procedure aligns with the training recipe outlined in Sec. 5.6, with our FiTv2-XL/2 model trained for $400K$ steps. Additionally, a baseline

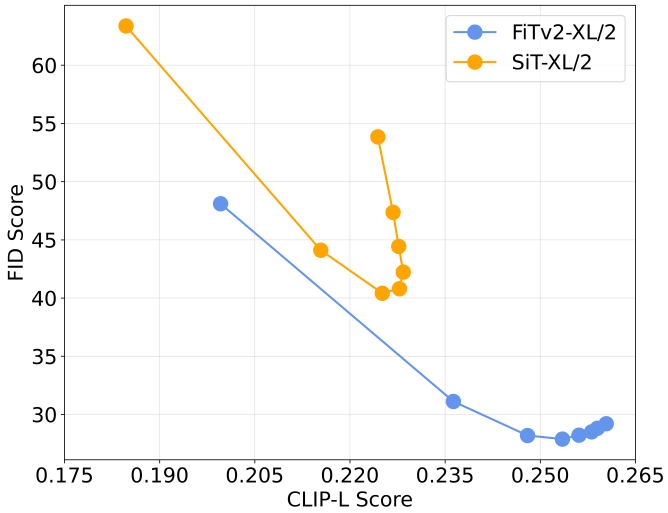


Fig. 10: **Comparision of FID and CLIP-L score across different CFG scales for two text-to-image models: FiTv2-XL/2 and SiT-XL/2.** FiTv2-XL/2 significantly outperforms SiT-XL/2 in terms of FID score and CLIP-L score.

SiT-XL/2 model is trained for the same $400K$ steps for comparative analysis. To ensure a fair comparison, SiT-XL/2 processes the text features and image latents in the same manner as our FiTv2, detailed in Sec. 4.3.4.

We evaluate our FiTv2-XL/2 and SiT-XL/2 for T2I generation on the standard MS-COCO benchmark at 256×256 resolution. Consistent with previous literature, we randomly sample $30K$ prompts from the MS-COCO validation set and generate images according to those prompts to compute the FID score and CLIP-L score. The Pareto curve is shown in Fig. 10 with classifier-free guidance factor of $[1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 9.0]$. With the same training steps, our FiTv2 achieves stronger results both on FID and CLIP scores, attaining an optimal FID of **27.88** and an optimal CLIP score of **0.2535** at cfg=4.0. In comparison, the SiT model reaches an optimal FID of **40.8** and an optimal CLIP score of **0.2278** at CFG=4.0. Combined with the qualitative results in Fig. 9, it is evident that our FiTv2 model beats the SiT model on T2I architecture.

6 Conclusion

In this work, we aim to contribute to the ongoing research on flexible generating arbitrary resolutions and aspect ratios. We propose a Flexible Vision Transformer (FiT) for the diffusion model, a refined transformer architecture with a flexible training pipeline specifically designed for generating images with arbitrary resolutions and aspect ratios. FiT and FiTv2 surpass all previous models, whether transformer-based or CNN-based, across various resolutions. With our resolution extrapolation method, Vision-NTK, and attention scale, the performance of FiTv2 has been significantly enhanced further. We also scale the FiTv2 to 3 billion to investigate the scalability of our model. Extensive experiments on class-guided image generation, flexible image generation, high-resolution image generation, and text-to-image generation demonstrate the effectiveness

of our FiTv2. We hope our work can inspire insights towards designing more powerful diffusion transformer models.

ACKNOWLEDGMENTS

This work is supported by the Shanghai Artificial Intelligence Laboratory.

REFERENCES

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *EEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009. 1, 2, 5, 9

[2] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *IEEE/CVF International Conference on Computer Vision*, 2023. 1, 2, 3, 5, 9, 13, 14, 15

[3] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, "Sdedit: Guided image synthesis and editing with stochastic differential equations," *arXiv preprint arXiv:2108.01073*, 2021. 1, 3, 7

[4] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, "Roformer: Enhanced transformer with rotary position embedding," *Neurocomputing*, 2024. 1, 3, 5

[5] X. Liu, H. Yan, S. Zhang, C. An, X. Qiu, and D. Lin, "Scaling laws of rope-based extrapolation," *arXiv preprint arXiv:2310.05209*, 2023. 1, 6

[6] N. Shazeer, "Glu variants improve transformer," *arXiv preprint arXiv:2002.05202*, 2020. 1, 3

[7] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022. 2, 3

[8] B. Peng, J. Quesnelle, H. Fan, and E. Shippole, "Yarn: Efficient context window extension of large language models," *arXiv preprint arXiv:2309.00071*, 2023. 2, 4, 5

[9] LocalLLaMA, "Ntk-aware scaled rope allows llama models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation," https://www.reddit.com/r/LocalLLaMA/comments/14lz7j5/ntkaware_scaled_rope_allows_llama_models_to_have/, accessed: 2024-2-1. 2, 4, 5

[10] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *EEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2, 3, 9, 13

[11] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in Neural Information Processing Systems*, 2021. 2, 7, 10, 13, 15

[12] N. Ma, M. Goldstein, M. S. Albergo, N. M. Boffi, E. Vanden-Eijnden, and S. Xie, "Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers," *arXiv preprint arXiv:2401.08740*, 2024. 2, 3, 7, 9, 13

[13] Z. Lu, Z. Wang, D. Huang, C. Wu, X. Liu, W. Ouyang, and L. Bai, "Fit: Flexible vision transformer for diffusion model," in *International Conference on Machine Learning*, 2024. 3

[14] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, 2020. 3, 4, 7

[15] X. Liu, C. Gong, and qiang liu, "Flow straight and fast: Learning to generate and transfer data with rectified flow," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=XVjTT1nw5z 3, 4, 7

[16] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *Advances in Neural Information Processing Systems*, 2022. 3

[17] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*, 2021. 3

[18] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, "Diffusion models in vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 9, pp. 10 850–10 869, 2023. 3

[19] S. Bond-Taylor, A. Leach, Y. Long, and C. G. Willcocks, "Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 11, pp. 7327–7347, 2021. 3

[20] A. Hyvärinen and P. Dayan, "Estimation of non-normalized statistical models by score matching." *Journal of Machine Learning Research*, 2005. 3

[21] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," *arXiv preprint arXiv:2011.13456*, 2020. 3, 4

[22] F. Ling, Z. Lu, J.-J. Luo, L. Bai, S. K. Behera, D. Jin, B. Pan, H. Jiang, and T. Yamagata, "Diffusion model-based probabilistic downscaling for 180-year east asian climate reconstruction," *npj Climate and Atmospheric Science*, 2024. 3

[23] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," in *EEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 3

[24] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," *arXiv preprint arXiv:2209.14988*, 2022. 3

[25] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020. 3, 4

[26] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013. 3

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017. 3

[28] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, 2020. 3

[29] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al.*, "Palm: Scaling language modeling with pathways," *Journal of Machine Learning Research*, 2023. 3

[30] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020. 3, 12

[31] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu *et al.*, "A survey on vision transformer," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 1, pp. 87–110, 2022. 3

[32] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth *et al.*, "Gemini: a family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023. 3

[33] H. Touvron, A. Vedaldi, M. Douze, and H. Jégou, "Fixing the train-test resolution discrepancy," *Advances in Neural Information Processing Systems*, 2019. 3

[34] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International Conference on Machine Learning*, 2021. 3

[35] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *EEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 3

[36] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *EEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 3

[37] M. Dehghani, B. Mustafa, J. Djolonga, J. Heek, M. Minderer, M. Caron, A. Steiner, J. Puigcerver, R. Geirhos, I. Alabdulmohsin *et al.*, "Patch n'pack: Navit, a vision transformer for any aspect ratio and resolution," *arXiv preprint arXiv:2307.06304*, 2023. 3

[38] S. Gao, P. Zhou, M.-M. Cheng, and S. Yan, "Masked diffusion transformer is a strong image synthesizer," *arXiv preprint arXiv:2303.14389*, 2023. 3, 13

[39] F. Bao, S. Nie, K. Xue, Y. Cao, C. Li, H. Su, and J. Zhu, "All are worth words: A vit backbone for diffusion models," in *EEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 3, 13, 15

[40] A. Hatamizadeh, J. Song, G. Liu, J. Kautz, and A. Vahdat, "Diffit: Diffusion vision transformers for image generation," *arXiv preprint arXiv:2312.02139*, 2023. 3, 15

[41] P. Gao, L. Zhuo, Z. Lin, C. Liu, J. Chen, R. Du, E. Xie, X. Luo, L. Qiu, Y. Zhang *et al.*, "Lumina-t2x: Transforming text into any modality, resolution, and duration via flow-based large diffusion transformers," *arXiv preprint arXiv:2405.05945*, 2024. 3, 13, 15

[42] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, and P. B. et al, "Palm: Scaling language modeling with pathways," *Journal of Machine Learning Research*, 2023. 3

[43] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, and B. R. et al, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023a. 3, 6

[44] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, and N. B. et al, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023b. 3, 6

[45] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang *et al.*, "Qwen technical report," *arXiv preprint arXiv:2309.16609*, 2023. 3

[46] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang *et al.*, "Qwen2 technical report," *arXiv preprint arXiv:2407.10671*, 2024. 3

[47] S. Chen, S. Wong, L. Chen, and Y. Tian, "Extending context window of large language models via positional interpolation," *arXiv preprint arXiv:2306.15595*, 2023. 4

[48] A. Ruoss, G. Delétang, T. Genewein, J. Grau-Moya, R. Csordás, M. Bennani, S. Legg, and J. Veness, "Randomized positional encodings boost length generalization of transformers," *arXiv preprint arXiv:2305.16843*, 2023. 4

[49] Y. Sun, L. Dong, B. Patra, S. Ma, S. Huang, A. Benhaim, V. Chaudhary, X. Song, , and F. Wei, "A length-extrapolatable transformer," *arXiv preprint arXiv:2212.10554*, 2022. 4

[50] Z. Jin, X. Shen, B. Li, and X. Xue, "Training-free diffusion model adaptation for variable-sized text-to-image synthesis," *Advances in Neural Information Processing Systems*, vol. 36, 2024. 4, 9

[51] J. Su, "Revisiting attention scale operation from the invariance of entropy," https://kexue.fm/archives/8823. 4, 9

[52] M. Dehghani, J. Djolonga, B. Mustafa, P. Padlewski, J. Heek, J. Gilmer, A. P. Steiner, M. Caron, R. Geirhos, I. Alabdulmohsin *et al.*, "Scaling vision transformers to 22 billion parameters," in *International Conference on Machine Learning*. PMLR, 2023, pp. 7480–7512. 7

[53] A. Gupta, L. Yu, K. Sohn, X. Gu, M. Hahn, L. Fei-Fei, I. Essa, L. Jiang, and J. Lezama, "Photorealistic video generation with diffusion models," *arXiv preprint arXiv:2312.06662*, 2023. 7

[54] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel *et al.*, "Scaling rectified flow transformers for high-resolution image synthesis," in *Forty-first International Conference on Machine Learning*, 2024. 7, 8

[55] C. Nash, J. Menick, S. Dieleman, and P. W. Battaglia, "Generating images with sparse representations," *arXiv preprint arXiv:2103.03841*, 2021. 7, 10

[56] T. Karras, M. Aittala, T. Aila, and S. Laine, "Elucidating the design space of diffusion-based generative models," *Advances in neural information processing systems*, vol. 35, pp. 26 565–26 577, 2022. 8

[57] E. B. Zaken, S. Ravfogel, and Y. Goldberg, "Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," *arXiv preprint arXiv:2106.10199*, 2021. 8

[58] G. Sun, W. Liang, J. Dong, J. Li, Z. Ding, and Y. Cong, "Create your world: Lifelong text-to-image diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 8

[59] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach, "Sdxl: Improving latent diffusion models for high-resolution image synthesis," *arXiv preprint arXiv:2307.01952*, 2023. 8, 15, 19

[60] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763. 8, 15, 19

[61] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017. 9

[62] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in Neural Information Processing Systems*, 2017. 10

[63] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in Neural Information Processing Systems*, 2016. 10

[64] T. Kynkäänniemi, T. Karras, S. Laine, and T. Lehtinen, J.and Aila, "Improved precision and recall metric for assessing generative models," *Advances in Neural Information Processing Systems*, 2019. 10

[65] J. Ho and T. Salimans, "Classifier-free diffusion guidance," in *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 10

[66] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," *arXiv preprint arXiv:1809.11096*, 2018. 13

[67] A. Sauer, K. Schwarz, and A. Geiger, "Stylegan-xl: Scaling stylegan to large diverse datasets," in *ACM SIGGRAPH 2022 conference proceedings*, 2022. 13

[68] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman, "Maskgit: Masked generative image transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 13, 15

[69] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans, "Cascaded diffusion models for high fidelity image generation," *Journal of Machine Learning Research*, 2022. 13

[70] H. Zheng, W. Nie, A. Vahdat, and A. Anandkumar, "Fast training of diffusion models with masked transformers," *Transactions on Machine Learning Research*, 2023. 13, 15

[71] Y. Pu, Z. Xia, J. Guo, D. Han, Q. Li, D. Li, Y. Yuan, J. Li, Y. Han, S. Song *et al.*, "Efficient diffusion transformer with step-wise dynamic attention mediators," *arXiv preprint arXiv:2408.05710*, 2024. 13

[72] E. Hoogeboom, J. Heek, and T. Salimans, "simple diffusion: End-to-end diffusion for high resolution images," in *International Conference on Machine Learning*. PMLR, 2023, pp. 13 213–13 232. 13, 15

[73] Y. Teng, Y. Wu, H. Shi, X. Ning, G. Dai, Y. Wang, Z. Li, and X. Liu, "Dim: Diffusion mamba for efficient high-resolution image synthesis," *arXiv preprint arXiv:2405.14224*, 2024. 15

[74] J. N. Yan, J. Gu, and A. M. Rush, "Diffusion models without attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 8239–8249. 15

[75] S. Changpinyo, P. Sharma, N. Ding, and R. Soricut, "Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 3558–3568. 15, 19

[76] V. Singla, K. Yue, S. Paul, R. Shirkavand, M. Jayawardhana, A. Ganjdanesh, H. Huang, A. Bhatele, G. Somepalli, and T. Goldstein, "From pixels to prose: A large dataset of dense image captions," *arXiv preprint arXiv:2406.10328*, 2024. 15

[77] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *Advances in neural information processing systems*, vol. 35, pp. 36 479–36 494, 2022. 15

[78] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*, 2014. 19

[79] T. Brooks, B. Peebles, C. Holmes, W. DePue, Y. Guo, L. Jing, D. Schnurr, J. Taylor, T. Luhman, E. Luhman, C. Ng, R. Wang, and A. Ramesh, "Video generation models as world simulators," https://openai.com/sora, 2024, accessed: 2024-5-1. 20

[80] Z. Wang, Z. Lu, D. Huang, T. He, X. Liu, W. Ouyang, and L. Bai, "Predbench: Benchmarking spatio-temporal prediction across diverse disciplines," *arXiv preprint arXiv:2407.08418*, 2024. 20

[81] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *EEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 20

[82] Z. Lu, J. Jiang, J. Huang, G. Wu, and X. Liu, "Glama: Joint spatial and frequency loss for general image inpainting," in *EEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 20

APPENDIX A
EXPERIMENTIN SETUPS

We provide detailed network configurations and performance of all models, which are listed in Tab. 8.

Models	Layers	Dim.	Head Num.	Patch Size	Max Token Length	Training Steps	Batch Size	Learning Rate	FID-50K
DiT-B/2	12	768	12	2	256	400K	256	1×10^{-4}	44.83
DiT-XL/2	28	1152	16	2	256	7000K	256	1×10^{-4}	9.62
DiT-XL-G/2	28	1152	16	2	256	7000K	256	1×10^{-4}	2.27
SiT-B/2	12	768	12	2	256	400K	256	1×10^{-4}	34.84
SiT-XL/2	28	1152	16	2	256	700K	256	1×10^{-4}	9.35
SiT-XL/2-G	28	1152	16	2	256	700K	256	1×10^{-4}	2.15
FiT Config A	12	768	12	2	256	400K	256	1×10^{-4}	43.34
FiT Config B	12	768	12	2	256	400K	256	1×10^{-4}	41.75
FiT Config C	12	768	12	2	256	400K	256	1×10^{-4}	39.11
FiT Config D	12	768	12	2	256	400K	256	1×10^{-4}	37.29
FiT-B/2	12	768	12	2	256	400K	256	1×10^{-4}	36.36
FiT-B/2	12	768	12	2	256	1500K	256	1×10^{-4}	26.08
FiT-XL/2	28	1152	16	2	256	2000K	256	1×10^{-4}	10.65
FiT-XL/2-G	28	1152	16	2	256	2000K	256	1×10^{-4}	4.21
FiT Config E	12	768	12	2	256	2000K	256	1×10^{-4}	21.23
FiT Config F	12	768	12	2	256	2000K	256	1×10^{-4}	20.73
FiT Config G	12	768	12	2	256	2000K	256	1×10^{-4}	18.42
FiT Config H	12	768	12	2	256	1500K	256	1×10^{-4}	22.04
FiT Config I	12	768	12	2	256	2000K	256	1×10^{-4}	22.04
FiT Config J	12	768	12	2	256	2000K	256	1×10^{-4}	19.21
FiTv2-B/2	12	768	12	2	256	400K	256	1×10^{-4}	26.03
FiTv2-B/2	12	768	12	2	256	2000K	256	1×10^{-4}	16.52
FiTv2-XL/2	28	1152	16	2	256	2000K	256	1×10^{-4}	9.24
FiTv2-XL/2-G	28	1152	16	2	256	2000K	256	1×10^{-4}	2.26
FiTv2-3B/2	28	1152	16	2	256	1000K	256	1×10^{-4}	7.49
FiTv2-3B/2-G	28	1152	16	2	256	1000K	256	1×10^{-4}	2.15

TABLE 8: Network configurations and performance of all models.

We use the same ft-EMA VAE[1] with DiT, which is provided by the Stable Diffusion to encode/decode the image/latent tokens by default. The metrics are calculated using the ADM TensorFlow evaluation Suite[2]. For DiT and FiT, we use DDPM sampler with 250 steps, while ODE sampler dopri5 [3] is adopted in SiT and FiTv2.

APPENDIX B
NETWORK FLOPS ANALYSIS

Models	Training Steps	FID	sFID	IS	Precision	Recall	Inference GFLOPs	Training GFLOPs ↑
FiTv2-B/2	200K	37.28	6.22	39.32	0.53	0.62	27.3	5460
FiTv2-B/2	400K	26.03	5.84	57.70	0.58	0.63	27.3	10920
FiTv2-B/2	1000K	19.03	5.64	76.81	0.62	0.64	27.3	27300
FiTv2-XL/2	200K	17.57	5.07	74.34	0.65	0.61	147	29400
FiTv2-B/2	1500K	17.71	5.64	81.57	0.63	0.65	27.3	40950
FiTv2-B/2	2000K	16.52	5.61	86.25	0.63	0.65	27.3	54600
FiTv2-XL/2	400K	11.75	4.75	100.54	0.68	0.64	147	58800
FiTv2-XL/2	700K	9.91	4.81	115.24	0.68	0.65	147	102900
FiTv2-3B/2	200K	10.74	4.95	102.88	0.70	062	653	130600
FiTv2-XL/2	1000K	9.30	4.91	120.78	0.68	0.66	147	147000
FiTv2-XL/2	1400K	9.29	5.03	123.43	0.67	0.67	147	205800
FiTv2-3B/2	400K	7.94	4.70	126.39	0.71	0.64	653	261200
FiTv2-XL/2	2000K	9.24	5.15	128.13	0.67	0.68	147	294000
FiTv2-3B/2	1000K	7.49	4.78	140.10	0.69	0.68	653	653000

TABLE 9: Network capacity, training FLOPs, inference FLOPs, and generation quality of all models.

We conduct a more comprehensive experiment for FiT to analyze the trade-offs between model capacity, training GFLOPs, inference GFLOPs, and generation quality, as shown in Tab. 9. We sort the tables according to training FLOPs

1. https://huggingface.co/stabilityai/sd-vae-ft-ema
2. https://github.com/openai/guided-diffusion/tree/main/evaluations
3. https://github.com/rtqichen/torchdiffeq

and we can find that: (1) Larger training GFLOPs can improve model performance: As Training FFLOPs are increased, and FID is decreased. These results indicate that scaling model training GFLOPs is the key to improved performance. (2) Larger model capacity under the same training steps can improve model performance: As model capacity is increased and training steps are held constant ($400K$), FID is decreased. These results indicate that scaling model capacity is the key to improved performance.

APPENDIX C
TEXT-TO-IMAGE EXPERIMENTS

We trained a text-to-image model on a larger and more complex dataset, CC12M [75], to evaluate the performance of FiTv2. In terms of architecture, we referenced the previous work design to employ text-image concatenation to inject text information. The hyperparameter configuration employed aligns with that of FiTv2-XL/2 and SiT-XL/2 in the ImageNet dataset. For the text encoder, we utilize the pre-trained CLIP-L [60] text encoder [4]. For the image encoder, we utilize the pre-trained VAE [5] from SDXL [59]. The evaluation of FiTv2-XL/2 and SiT-XL/2 models was conducted at 400K training steps using FID-30K and CLIP score on MSCOCO [78], as shown in Tabs. 10 and 11.

FID	CFG=1.0	CFG=2.0	CFG=3.0	CFG=4.0	CFG=5.0	CFG=6.0	CFG=7.0	CFG=9.0
FiTv2-XL/2	48.1	31.12	28.19	27.88	28.22	28.51	28.81	29.2
SiT-XL/2	63.37	44.11	40.41	40.8	42.23	44.43	47.36	53.85

TABLE 10: FID performance of FiT-XL/2 model with different classifier-free-guidance at 256×256 resolution on MS-COCO-30K benchmark.

CLIP	CFG=1.0	CFG=2.0	CFG=3.0	CFG=4.0	CFG=5.0	CFG=6.0	CFG=7.0	CFG=9.0
FiTv2-XL/2	0.1996	0.2363	0.248	0.2535	0.2561	0.2581	0.259	0.2604
SiT-XL/2	0.1847	0.2154	0.2251	0.2278	0.2284	0.2277	0.2268	0.2244

TABLE 11: CLIP performance of FiT-XL/2 model with different classifier-free-guidance at 256×256 resolution on MS-COCO-30K benchmark.

APPENDIX D
DETAILED ATTENTION SCORE WITH 2D ROPE AND DECOUPLED 2D-ROPE.

2D RoPE defines a vector-valued complex function $f(\mathbf{x}, h_m, w_m)$ in Eq. (20) as follows:

$$\begin{aligned} f(\mathbf{x}, h_m, w_m) = \Big[&(x_0 + ix_1)e^{ih_m\theta_0}, (x_2 + ix_3)e^{ih_m\theta_1}, \dots, (x_{d/2-2} + ix_{d/2-1})e^{ih_m\theta_{d/4-1}}, \\ &(x_{d/2} + ix_{d/2+1})e^{iw_m\theta_0}, (x_{d/2+2} + ix_{d/2+3})e^{iw_m\theta_1}, \dots, (x_{d-2} + ix_{d-1})e^{iw_m\theta_{d/4-1}}\Big]^T. \end{aligned} \tag{33}$$

The self-attention score A_n injected with 2D RoPE in Eq. (21) is detailed defined as follows:

$$\begin{aligned} A_n =& \mathrm{Re}\langle f_q(\mathbf{q}_m, h_m, w_m), f_k(\mathbf{k}_n, h_n, w_n)\rangle \\ =& \mathrm{Re}\left[\sum_{j=0}^{d/4-1}(q_{2j} + iq_{2j+1})(k_{2j} - ik_{2j+1})e^{i(h_m-h_n)\theta_j} + \sum_{j=0}^{d/4-1}(q_{2j} + iq_{2j+1})(k_{2j} - ik_{2j+1})e^{i(w_m-w_n)\theta_j}\right] \\ =& \sum_{j=0}^{d/4-1}[(q_{2j}k_{2j} + q_{2j+1}k_{2j+1})\cos((h_m - h_n)\theta_j) + (q_{2j}k_{2j+1} - q_{2j+1}k_{2j})\sin((h_m - h_n)\theta_j)] + \\ & \sum_{j=0}^{d/4-1}[(q_{2j}k_{2j} + q_{2j+1}k_{2j+1})\cos((w_m - w_n)\theta_j) + (q_{2j}k_{2j+1} - q_{2j+1}k_{2j})\sin((w_m - w_n)\theta_j)], \end{aligned} \tag{34}$$

where 2-D coordinates of width and height as $\{(w,h)\big|1 \leqslant w \leqslant W, 1 \leqslant h \leqslant H\}$, the subscripts of q and k denote the dimensions of the attention head, $\theta^n = 10000^{-2n/d}$. There is no cross-term between h and w in 2D-RoPE and attention score A_n, so we can further decouple the rotary frequency as $\Theta_h = \{\theta_d^h = b_h^{-2d/|D|}, 1 \leqslant d \leqslant \frac{|D|}{2}\}$ and $\Theta_w = \{\theta_d^w = b_w^{-2d/|D|}, 1 \leqslant d \leqslant \frac{|D|}{2}\}$, resulting in the decoupled 2D-RoPE, as follows:

4. https://huggingface.co/openai/clip-vit-large-patch14
5. https://huggingface.co/stabilityai/sdxl-vae

$$
\begin{aligned}
A_n = & \sum_{j=0}^{d/4-1} [(q_{2j}k_{2j} + q_{2j+1}k_{2j+1})\cos((h_m - h_n)\theta_j^h) + (q_{2j}k_{2j+1} - q_{2j+1}k_{2j})\sin((h_m - h_n)\theta_j^h)] + \\
& \sum_{j=0}^{d/4-1} [(q_{2j}k_{2j} + q_{2j+1}k_{2j+1})\cos((w_m - w_n)\theta_j^w) + (q_{2j}k_{2j+1} - q_{2j+1}k_{2j})\sin((w_m - w_n)\theta_j^w)] \\
= & \mathrm{Re}\left[\sum_{j=0}^{d/4-1} (q_{2j} + iq_{2j+1})(k_{2j} - ik_{2j+1})e^{i(h_m - h_n)\theta_j^h} + \sum_{j=0}^{d/4-1} (q_{2j} + iq_{2j+1})(k_{2j} - ik_{2j+1})e^{i(w_m - w_n)\theta_j^w} \right] \\
= & \mathrm{Re}\langle \hat{f}_q(\mathbf{q}_m, h_m, w_m), \hat{f}_k(\mathbf{k}_n, h_n, w_n) \rangle.
\end{aligned}
\tag{35}
$$

So we can reformulate the vector-valued complex function $\hat{f}(\mathbf{x}, h_m, w_m)$ in Eq. (28) as follows:

$$
\begin{aligned}
\hat{f}(\mathbf{x}, h_m, w_m) = \Big[& (x_0 + ix_1)e^{ih_m\theta_0^h}, (x_2 + ix_3)e^{ih_m\theta_1^h}, \ldots, (x_{d/2-2} + ix_{d/2-1})e^{ih_m\theta_{d/4-1}^h}, \\
& (x_{d/2} + ix_{d/2+1})e^{iw_m\theta_0^w}, (x_{d/2+2} + ix_{d/2+3})e^{iw_m\theta_1^w}, \ldots, (x_{d-2} + ix_{d-1})e^{iw_m\theta_{d/4-1}^w} \Big]^T .
\end{aligned}
\tag{36}
$$

APPENDIX E
MORE MODEL SAMPLES

We show samples from our FiTv2-3B/2 models at resolutions of 512×512, 256×768, and 768×256, trained for $1000K$. All the images are sampled with CFG=4.0, see Figs. 11 to 16 for detials. More T2I results of FiTv2-XL/2 at $400K$ taining steps and 256×256 resolution are shown in Fig. 17

We also show some failure samples from DiT-XL/2, as shown in Fig. 18. These samples illustrate two typical failure modes of DiT: (1) Synthesized objects can be cropped, such as the cut-off head of the elephant in the examples. (2) Synthesized images are blurry, such as the dogs in the examples are very blurry and accompanied by various artifacts. An intuitive explanation for these failures is the use of random cropping and resizing during training of the model: In Deep Learning frameworks like PyTorch, the aggregation of a batch necessitates tensors of identical dimensions. Consequently, a typical processing pipeline is to resize an image such that the shortest size matches the desired target size, followed by randomly cropping the image along the longer axis. While random cropping and resizing are natural forms of data augmentation, they can leak into the generated samples, causing the malicious effects shown in Fig. 18.

APPENDIX F
LIMITATIONS AND FUTURE WORK

Although FiTv2 has demonstrated outstanding performance in the field of image generation, it still has certain limitations. We plan to address these limitations in our future work:

1. Our model training and evaluation have been constrained to the token length of 256 and 1024. In future work, we plan to explore training the model to higher resolutions (longer maximum token lengths), such as investigating the performance of FiTv2 at 4K-resolution image generation.

2. We have only conducted the text-to-image (T2I) generation experiment of FiTv2 on limited datasets with limited training steps. We plan to explore the scalability of FiTv2 on T2I tasks.

3. The current study has only focused on the imaging modality of FiT. In future work, we will explore FiTv2's capabilities in other modalities, such as video generation [79], [80], and other applications, such as image inpainting [81], [82]. Leveraging the architectural design of the FiTv2, we can generate videos with flexible resolution and frame rates.

Fig. 11: Uncurated samples from FiTv2-3B/2 models at resolutions of 512×512, 256×768 and 768×256.

Fig. 12: Uncurated samples from FiTv2-3B/2 models at resolutions of 512×512, 256×768 and 768×256.

Fig. 13: Uncurated samples from FiTv2-3B/2 models at resolutions of 512×512, 256×768 and 768×256.

Fig. 14: Uncurated samples from FiTv2-3B/2 models at resolutions of 512×512, 256×768 and 768×256.

Fig. 15: Uncurated samples from FiTv2-3B/2 models at resolutions of 512×512, 256×768 and 768×256.

Fig. 16: Uncurated samples from FiTv2-3B/2 models at resolutions of 512×512, 256×768 and 768×256.

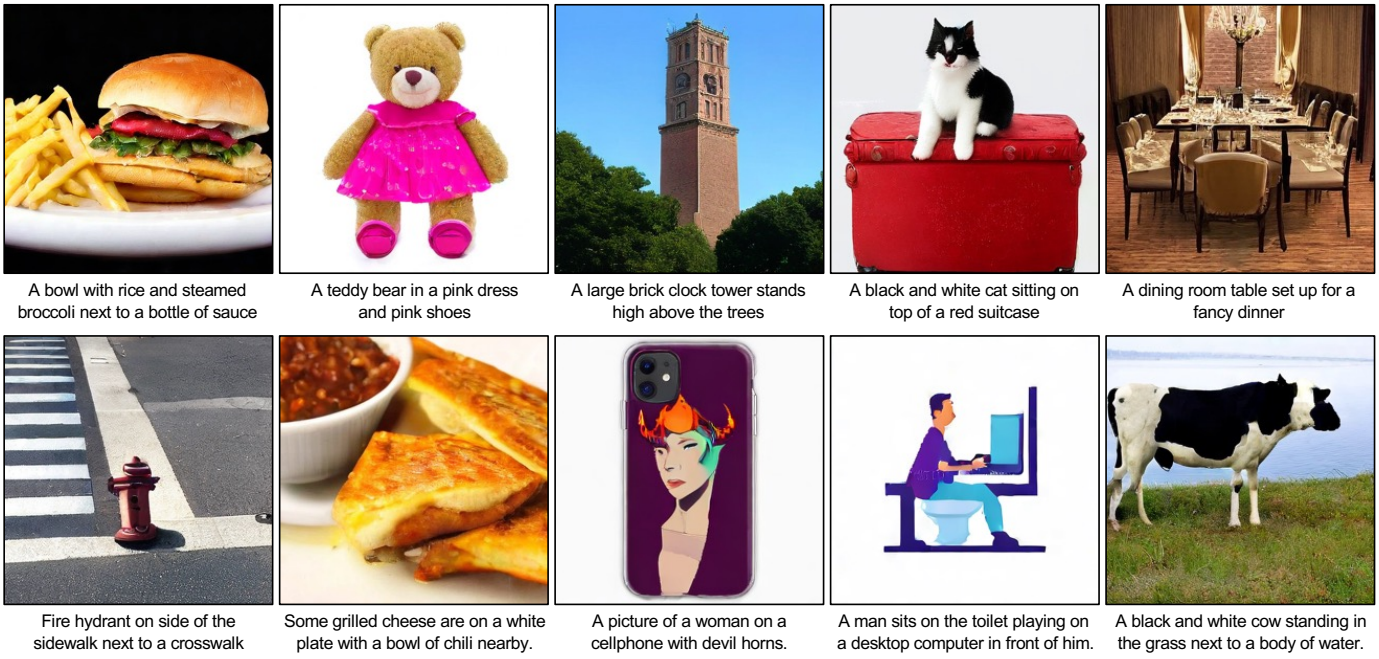


Fig. 17: **Uncurated samples from FiTv2-XL/2 models at resolutions of** 256×256 **on text-to-image generation tasks.** All the images are sampled with CFG=4.0. With only $400K$ training steps, our model is capable of generating releastic images according to text descriptions.

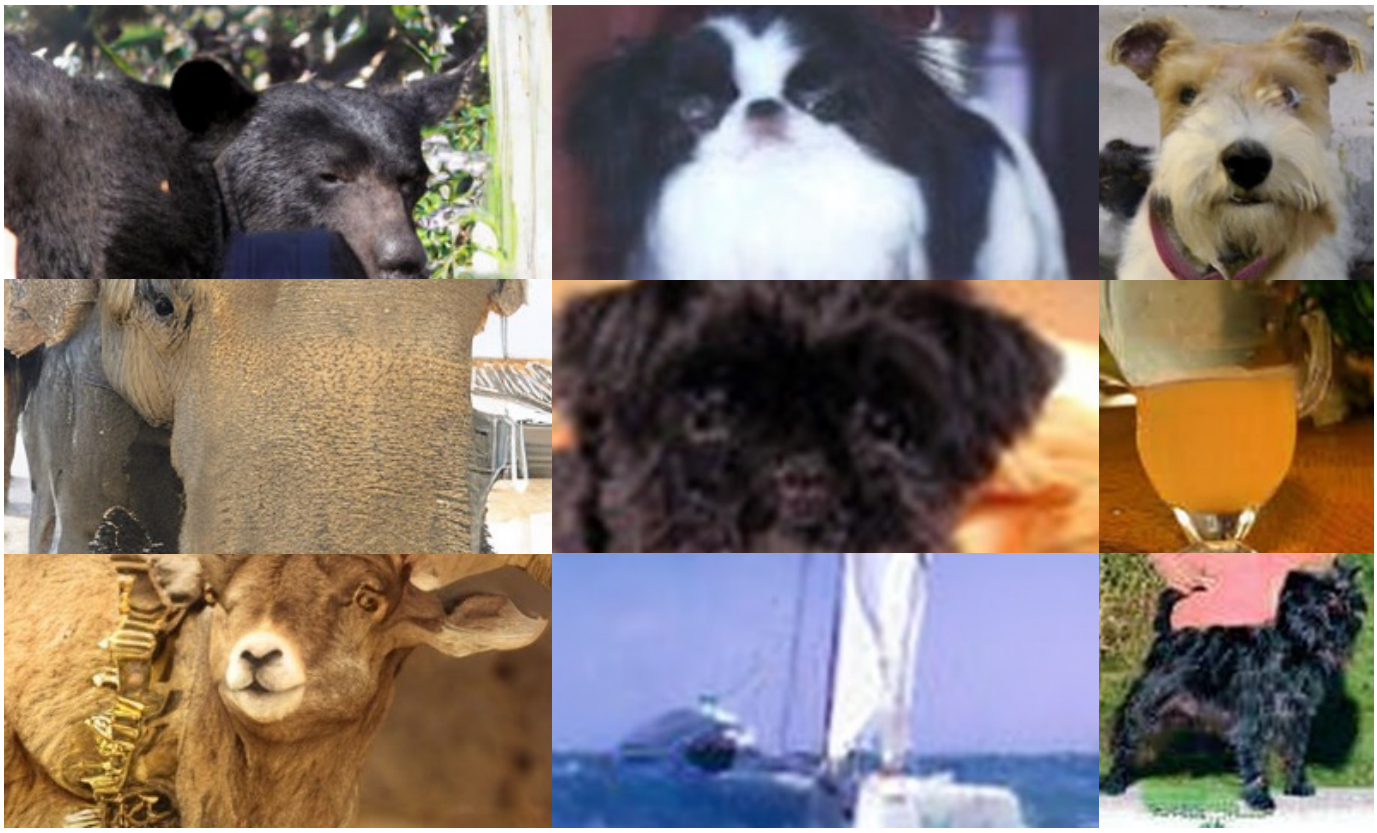


Fig. 18: Uncurated failure samples from DiT-XL/2.