Native-Resolution Image Synthesis

Zidong Wang[1,2], **Lei Bai**[2,*], **Xiangyu Yue**[1], **Wanli Ouyang**[1,2], **Yiyuan Zhang**[1,2 *]
[1]MMLab, CUHK [2]Shanghai AI Lab
wangzd2022@gmail.com, xyyue@cuhk.edu.hk, ouyangwanli@pjlab.org.cn
Codes and models: https://github.com/WZDTHU/NiT

Figure 1: **Native-resolution image synthesis on *ImageNet*.** A single Native-resolution diffusion Transformer (NiT) model, trained on *ImageNet*, generates images across diverse, arbitrary resolutions and aspect ratios (examples shown from 256×256 to 2048×2048, and aspect ratios from $1:5$ to $3:1$). This capability extends far beyond conventional fixed-resolution, square-image generation (*e.g.*, 256×256), demonstrating strong generalization.

Abstract

We introduce native-resolution image synthesis, a novel generative modeling paradigm capable of synthesizing images at arbitrary resolutions and aspect ratios. This approach overcomes the limitations of conventional fixed-resolution, square-image methods by natively handling variable-length visual tokens—a core

*Corresponding authors: yiyuan@link.cuhk.edu.hk, bailei@pjlab.org.cn

Preprint. Under review.

challenge for traditional techniques. To this end, we introduce the Native-resolution diffusion Transformer (NiT), an architecture designed to explicitly model varying resolutions and aspect ratios within its denoising process. Free from the constraints of fixed formats, NiT learns intrinsic visual distributions from images spanning a broad range of resolutions and aspect ratios. Notably, a single NiT model simultaneously achieves the state-of-the-art performance on both *ImageNet*-256×256 and 512×512 benchmarks. Surprisingly, akin to the robust zero-shot capabilities seen in advanced Large Language Models, NiT, trained solely on ImageNet, demonstrates excellent zero-shot generalization performance. It successfully generates high-fidelity images at previously unseen high resolutions (e.g., 1536×1536) and diverse aspect ratios (e.g., $16:9, 3:1, 4:3$), as shown in Figure 1. These findings indicate the significant potential of native-resolution modeling as a bridge between visual generative modeling and advanced LLM methodologies.

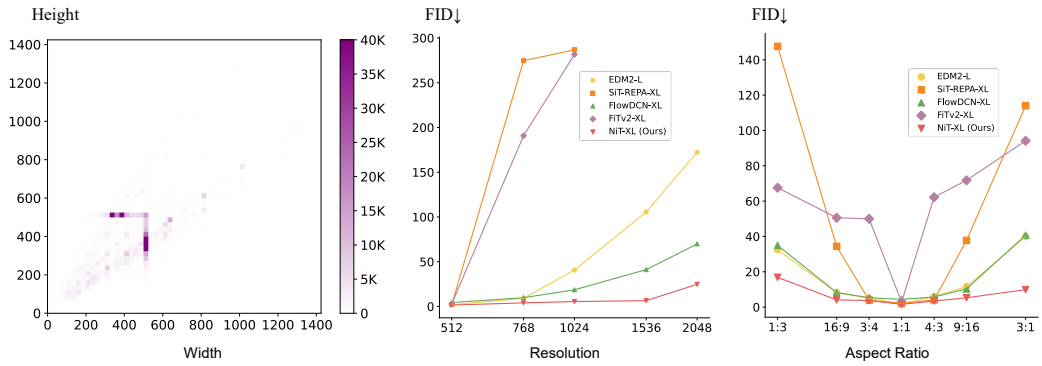
1 Introduction

The emergence of Large Language Models (LLMs) [1, 6, 22, 26, 44, 50, 74, 77, 78, 82] represents a transformative development in the AI-Generated Content (AIGC) area. Their success is largely attributed to two key characteristics: remarkable scalability and profound zero-shot generalization. Scalability is empirically validated by the established scaling laws [1, 35], which show predictable performance gains with increased model and data size. Generalization, meanwhile, is demonstrated by their capability to perform tasks for which they were not trained, such as seamlessly handling unseen questions of variable lengths, affording exceptional flexibility.

Concurrently, diffusion models [23, 31, 36, 37, 43, 48, 56, 58, 62, 67, 83, 86] have risen to prominence in visual generative modeling, lauded for their capacity to synthesize high-fidelity data. However, prevailing diffusion transformers [23, 48, 56] typically standardize images to fixed, often square, dimensions during training. This preprocessing step, while simplifying model architecture, inherently discards crucial native resolution and aspect ratio information. Such a practice curtails the models' ability to learn visual features across diverse scales and orientations [11, 47, 58, 79, 83], thereby limiting their intrinsic flexibility and generalization capabilities concerning input variability.

Large Language Models (LLMs) effectively process variable-length text by training directly on native data formats [1, 16, 22, 71, 72, 78, 82]. This inherent adaptability inspires a critical question for image synthesis: *Can diffusion models achieve similar flexibility, learning to generate images directly at their diverse, native resolutions and aspect ratios*? Conventional diffusion models exhibit significant challenges in generalizing across resolutions beyond their training regime. This limitation stems from three core difficulties: **1**) *Strong coupling between fixed receptive fields in convolutional architectures and learned feature scales* [18, 36, 37, 62]. The models internalize visual concepts at a resolution-specific scale, hindering effective feature extraction when resolution changes. **2**) *Fragility of positional encoding and spatial coordinate dependencies in transformer architectures* [25, 48, 56]. Hardcoded or learned positional encoding for a specific grid size, leads to distorted spatial reasoning and object coherence at novel resolutions. **3**) *Inefficient and Unstable Training Dynamics from Variable Inputs.* Padding variable inputs [47, 80] causes waste and artifacts; while aspect ratio bucketing [11, 58, 81] increases training complexity. Both methods harm efficiency and the diffusion process's sensitivity to resolution-dependent image statistics. Addressing these interconnected challenges is crucial for developing truly native-resolution diffusion models.

In this work, we overcome these limitations by proposing a novel architecture for diffusion transformers that directly models native-resolution image data for generation. Drawing inspiration from the variable-sequence nature of Vision Transformers [16, 21], we reformulate image generative modeling within diffusion transformers as "native-resolution generation". And we present the Native-resolution diffusion Transformer (NiT), which demonstrates the capability to generate images across a wide spectrum of resolutions and aspect ratios. By exclusively training on images at their original resolutions, without resolution-modifying augmentations, our model inherently captures robust spatial relationships. NiT is developed based on the DiT [56] architecture and incorporates several key architectural innovations: **1**) *Dynamic Tokenization* converts images in native resolution into variable-length token sequences and the tuples of corresponding height and width. Without requiring input padding, it avoids substantial computational overhead. **2**) *Variable-Length Sequence Processing*. We use Flash Attention [15] to natively process heterogeneous, unpadded token sequences by cumulative



(a) Data Distribution of ImageNet. (b) Resolution Generalization. (c) Aspect Ratio Generalization.

Figure 2: **NiT's Superior Generalization Beyond *ImageNet*'s Typical Resolution Distribution.** (a) *ImageNet* resolutions are mainly concentrated between 200 to 600 pixels (width/height), with sparse data beyond 800 pixels. Despite this, (b) shows our NiT model's superior generalization to unseen high resolutions (*e.g.*, 1024, 1536), evidenced by significantly lower FID scores. (c) further confirms NiT also exhibits the strongest generalization across various aspect ratios.

sequence lengths using the memory tiling strategy. **3**) *2D Structural Prior Injection*. We introduce the axial 2D Rotary Positional Embedding [69] (2D RoPE) to factorize the height and width impact and maximize the 2D structural prior by relative positional encoding.

Extensive experiments in class-guided image generation validate NiT as a significant advancement due to its native-resolution modeling. With **a single model**, NiT firstly attain state-of-the-art (SOTA) results on both 256×256 (2.08 FID, Fréchet inception distance) and 512×512 (**1.48** FID) benchmarks in class-guided *ImageNet* generation [17]. Impressively, NiT highlights its strong zero-shot generalization ability. For instance, as shown in Fig. 2, it achieves an FID of 4.52 on unseen 1024×1024 resolution and an FID of 4.11 on novel $9:16$ aspect ratio images (*i.e.*, 432×768), excelling in its flexibility and transferability to unfamiliar resolutions and respective ratios.

2 Related Work

2.1 Explorations of Variable-length Generalization

Large language models (LLMs) [1, 22, 50, 74, 77, 78, 82] are trained using text sequences with native, original length, which enables them to take flexible-length texts as input and generate arbitrary-length outputs. Seminal works, such as RoPE [69], NoPE [38], Alibi [59], and KERPLE [14], study the impact of positional encoding on the length generalization in language models. RoPE is then utilized in a wide range of LLMs as it unifies the relative position encoding with the absolute positional encoding. Beyond the valuable properties of RoPE, a series of works have been proposed to study the extreme context-length generalization of LLMs. NTK-RoPE [46], YaRN [57], LongRoPE [20], LM-Infinite [28], PoSE [87], and CLEX [9] explore the extreme length generalization of LLMs, enabling LLMs with the capability of a very long context length ($> 128K$) generation.

In the realm of computer vision, NaViT [16], DINO-v2 [55], and RADIO-v2.5 [29] have explored multi-resolution training to obtain a more robust vision representation. Recently, advanced vision language models (VLMs), including Qwen2-VL [4], Gemini1.5 [72], Intern-VL-2.5 [13], and Seed1.5-VL [27], have explored the native-resolution training in their vision encoders. However, the power of native-resolution training in visual content generation appears to be somewhat **locked**. In this work, we bridge the gap by exploring the native-resolution training in diffusion transformer models.

2.2 Explorations of Resolution Generalization in Visual Content Generation

Current visual generative models, whether autoregressive or diffusion-based, typically do not directly process visual content at its native resolution. Existing strategies to accommodate variable resolutions can be broadly categorized into three main approaches:

Bucket Sampling [11, 58, 81] facilitates dynamic resolution handling across different training batches by grouping samples into pre-defined "buckets". While the resolution and aspect ratio are fixed within each batch, they can vary between batches. The primary limitation is its reliance on these pre-defined buckets, restricting true flexibility to a discrete set of image dimensions.

Padding and Masking [47, 80] establishes a maximum sequence length, padding all image tokens to this length, and employing a mask to exclude padded regions from the loss calculation. This allows for dynamic resolution processing within a single batch. However, the approach often leads to significant computational and memory inefficiencies due to the processing of extensive padded areas, especially for images considerably smaller than the maximum resolution.

Progressive Multi-Resolution [10, 83] methods split the training process into several stages and progressively increase the image resolutions at each stage. While this method can effectively achieve high-resolution generation, it often exhibits suboptimal performance on smaller resolutions at earlier stages. Besides, it cannot generalize to higher resolutions beyond the last training stage.

The success of LLMs with variable-length inputs underscores a gap in visual content generation, where true native-resolution flexibility is still elusive. Existing methods to handle varied resolutions often introduce trade-offs in efficiency or generalization. To bridge this, we explore native-resolution training for diffusion transformer models, seeking a more fundamental solution to these persistent challenges. Our proposed methodology is detailed next.

3 Native-Resolution Diffusion Transformer

3.1 Preliminaries

Diffusion Foundation. Given the noise distribution $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ and the data distribution $x \sim p_{\text{data}}(x)$, the time-dependent forward diffusion process is defined as: $x_t = \alpha_t x + \sigma_t \epsilon$, where α_t is a decreasing function of t and σ_t is an increasing function of t. There are different strategies to train a diffusion model, including DDPM [31, 53], score matching [33, 65, 66], EDM [36, 37], and flow matching [2, 3, 43, 45]. We adopt flow matching with linear path in NiT, which restricts the forward and reverse process on $t \in [0, 1]$ and set $\alpha_t = 1 - t, \sigma_t = t$, interpolating between the two distributions with velocity target $v = \epsilon - x$. The Logit-Normal time distribution $t = \frac{\sigma}{1+\sigma}$ from EDM is introduced, where $\ln(\sigma) \sim \mathcal{N}(P_{\text{mean}}, P_{\text{std}}^2)$ with manually selected coefficients P_{mean} and P_{std}.

Conventional Fixed-Resolution Modeling The prevalent training strategy of diffusion models involves pre-setting a batch-wide image resolution, denoted as $H_{target} \times W_{target}$, often with $H_{target} = W_{target}$ for benchmarks like *ImageNet*-256×256. Images I_{orig} of native dimensions $H_{orig} \times W_{orig}$ are then subjected to resizing and cropping operations to conform. While simplifying model design, this fixed-resolution approach introduces three significant issues:

- *Spatial Structure and Semantic Degradation.* As discussed in previous works [34, 54, 68], resizing an image to a certain size via an interpolation function $f_{interp}(\cdot)$ with scaling factors s_H, s_W can be detrimental. Upsampling (if $s_H > 1$ or $s_W > 1$) often introduces blurriness, diminishing sharpness. Conversely, downsampling (if $s_H < 1$ or $s_W < 1$) leads to an irrecoverable loss of high-frequency details. Subsequent cropping to $H_{target} \times W_{target}$, particularly when aspect ratios $\frac{W_{orig}}{H_{orig}} \neq \frac{W_{target}}{H_{target}}$, discards image regions, potentially leading to semantic incompleteness or contextual loss, an effect observed to influence generated samples in models, as revealed in SDXL [58].
- *Inhibited Resolution Generalization.* Models trained exclusively at a fixed resolution (H_{target}, W_{target}) exhibit poor generative performance at novel resolutions $(H_{infer}, W_{infer}) \neq (H_{target}, W_{target})$.[2] This limitation is particularly acute for Transformer-based diffusion models, like DiT [56] and SiT [48]. Their reliance on absolute positional embeddings lacks 2D structural modeling, which does not readily adapt to changes in the number or spatial arrangement of patches resulting from differing resolutions.
- *Inefficient Data Utilization and Computational Overhead.* Natural image datasets contain rich visual information encoded in their diverse native resolutions. Standardizing all inputs to

[2]We provide a detailed comparison on this generalization ability in Table 2, and 3.

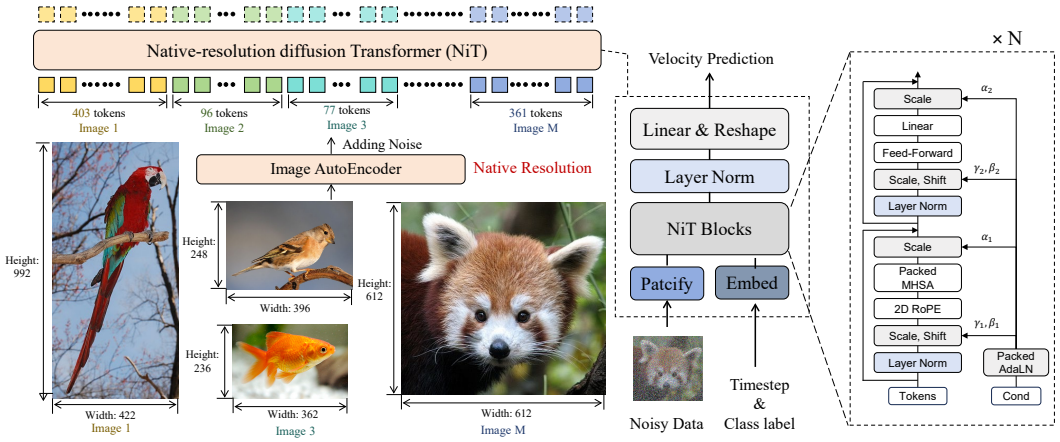


Figure 3: **Architecture Design of Native Resolution Diffusion Transformer (NiT).** NiT takes noisy latent representations, tokenizes them into variable-length sequences based on the original image resolution. Each NiT block utilizes Packed Multi-Head-Self-Attention (MHSA) with 2D RoPE and incorporates timestep and class conditioning via adaptive layer normalization.

$H_{target} \times W_{target}$ discards this inherent scale diversity [58]. Consequently, achieving high performance across multiple resolutions necessitates training distinct model instances for each specific resolution. So, for different resolutions, current approaches incur a cumulative computational training cost.[3]

3.2 Native-Resolution Modeling

Reformulation. As illustrated in Fig. 3, given a sequence of images in arbitrary resolutions, we use the image autoencoder to compress the image sequence to a latent sequence $\boldsymbol{x} = \{x_1, x_2, x_3, \cdots, x_n\}$. Similar to LLMs [4], we pre-define a maximum sequence length L and pack the latents $x_i \in \mathbb{R}^{c \times h_i \times w_i}$ together, where c, h_i and w_i are the dimension, height and width of the i-th latent, and p is the patch size. Please note that in the packing algorithm, to maintain the maximum sequence length, image instance number n is also dynamic in different iterations.

$$\begin{aligned} x_i \in \mathbb{R}^{c \times h_i \times w_i} &\Rightarrow x_i \in \mathbb{R}^{(\frac{h_i \cdot w_i}{p^2}) \times (c \cdot p \cdot p)}, \\ \boldsymbol{x} = \{x_1, x_2, x_3, \cdots, x_n\} &\Rightarrow \boldsymbol{x} \in \mathbb{R}^{(\frac{1}{p^2}\Sigma_i h_i \cdot w_i) \times (c \cdot p \cdot p)}, \end{aligned} \tag{1}$$

where the packing operation is to concatenate the variable-length latents to the maximum sequence length and improve computation efficiency.

After packing, we add noise and patchify the packed latent sequence $\boldsymbol{x}$. For each latent x_i, we sample each Gaussian noise in a variable-length independently, where the noise $\epsilon_i \in \mathbb{R}^{(\frac{h_i \cdot w_i}{p^2}) \times (c \cdot p \cdot p)}$ and time $t_i \in [0, 1]$ are added to the real data x_i:

$$x_i' = (1 - t_i) \cdot x_i + t_i \cdot \epsilon_i, \tag{2}$$

Then we use the patch embedding layers to project the noisy data to the visual tokens $\boldsymbol{z}$:

$$\mathbf{z}_0 = \texttt{PatchEmbed}(\boldsymbol{x}) \in \mathbb{R}^{(\frac{1}{p^2}\Sigma_i h_i \cdot w_i) \times d}, \tag{3}$$

where d is the hidden size of the diffusion transformer. Thus, a sequence of native-resolution images is formulated into a packed sequence of visual tokens, and it satisfies: $(\frac{1}{p^2}\Sigma_i h_i \cdot w_i) \leqslant L$.

[3] For example, despite SiT-REPA [86] achieving strong results on ImageNet-256 $\times$ 256, a separate model, and thus roughly 2$\times$ the training compute, are required for 512 $\times$ 512 resolution, highlighting an inefficient use of resources compared to models capable of handling variable resolutions.

[4] We adopt an efficient longest-pack-first histogram packing algorithm [40].

Use axial 2D-RoPE to inject 2D structural priors. The advantages of 2D Rotary Position Embedding (RoPE) for modeling inputs at their native resolutions are two-fold: **1**) 2D RoPE explicitly models 2D structural relationships within the image plane, offering better adaptability to various resolutions compared to learnable positional embeddings. **2**) The axial nature of 2D RoPE decouples height and width modeling, independently generating 1D rotational frequency components based on each token's patchfied height h'_k and width w'_k. Let $\mathbf{z}_k$ be the tokens associated with height-width coordinates (h'_k, w'_k). d is the hidden size of query ($\mathbf{q}_k$) and key ($\mathbf{k}_k$) vectors derived from $\mathbf{z}_k$. The dimensionality of the rotary angle space is $d_s = d/2$. Base angular frequencies ω_j are defined as:

$$\omega_j = \theta^{-2j/d_s} \quad \text{for } j \in \{0, \ldots, d_s/2 - 1\}, \tag{4}$$

where θ is a hyperparameter.[5] The composite angle vector $\mathbf{\Phi}_{h'_k, w'_k} \in \mathbb{R}^{d_s}$ for the token at position (h'_k, w'_k) is formed by concatenating angles derived from its height and width:

$$\mathbf{\Phi}_{h'_k, w'_k} = \texttt{Concat}\left(\{h'_k \cdot \omega_j\}_{j=0}^{d_s/2-1}, \{w'_k \cdot \omega_j\}_{j=0}^{d_s/2-1}\right). \tag{5}$$

In the self-attention mechanism, the query $\mathbf{q}_k \in \mathbb{R}^d$ and key $\mathbf{k}_k \in \mathbb{R}^d$ are transformed using these rotary embeddings. For any such vector $\mathbf{v} \in \mathbb{R}^d$ (representing either $\mathbf{q}_k$ or $\mathbf{k}_k$), its transformed version $\mathbf{v}'$ is obtained by rotating its feature pairs. Specifically, for each $l \in \{0, \ldots, d_s - 1\}$:

$$\begin{pmatrix} v'_{2l} \\ v'_{2l+1} \end{pmatrix} = \begin{pmatrix} \cos\left((\mathbf{\Phi}_{h'_k, w'_k})_l\right) & -\sin\left((\mathbf{\Phi}_{h'_k, w'_k})_l\right) \\ \sin\left((\mathbf{\Phi}_{h'_k, w'_k})_l\right) & \cos\left((\mathbf{\Phi}_{h'_k, w'_k})_l\right) \end{pmatrix} \begin{pmatrix} v_{2l} \\ v_{2l+1} \end{pmatrix}. \tag{6}$$

This axial 2D RoPE effectively injects 2D structural priors suitable for variable heights and widths, while inherently preserving RoPE's capacity to encode relative positions within the self-attention.

Packed Full-Attention. Processing packed sequences, which concatenate multiple variable-length visual token sequences from distinct data instances, necessitates restricting self-attention computations to operate only *within* the tokens of each original instance. While standard attention masking can enforce this, its application to the highly sparse structure of packed sequences incurs prohibitive computational overhead. Drawing inspiration from the efficient native-resolution processing in advanced Vision Language Models (VLMs) [4, 27, 72, 75], we employ FlashAttention-2 [15] to achieve this efficiently. Specifically, for a packed batch comprising n data instances, where the i-th instance contributes $N_i = (h_i \cdot w_i)/p^2$ tokens (derived from its latent dimensions h_i, w_i and patch size p), we define the individual token sequence lengths and their cumulative sums:

$$\texttt{CuSeqLens} = [0, N_1, N_1 + N_2, \ldots, \sum_{j=1}^{n-1} N_j, \sum_{j=1}^{n} N_j]. \tag{7}$$

This leverages FlashAttention-2's ability to handle batched inputs with variable sequence lengths (specified by `CuSeqLens`), thereby enabling efficient full-attention within each data instance without explicit masking or padding. A detailed algorithm is demonstrated in Algorithm 1

Packed Adaptive Layer Normalization. Conventional Adaptive Layer Normalization (AdaLN) methods are not directly suited for packed sequences due to the heterogeneity in sequence lengths and the need for instance-specific conditioning. To address this, we introduce Packed Adaptive Layer Normalization. For each k-th data instance within the packed sequence, its unique conditional embedding $\mathbf{c}_k \in \mathbb{R}^d$ (where d is the token feature dimension) is utilized to modulate its corresponding visual tokens. Specifically, $\mathbf{c}_k$ is first projected to produce instance-specific scale ($\boldsymbol{\gamma}_k \in \mathbb{R}^d$) and shift ($\boldsymbol{\beta}_k \in \mathbb{R}^d$) parameters. These parameters are then broadcast across all $N_k = (h_k \cdot w_k)/p^2$ tokens originating from the k-th data instance. If $\hat{\boldsymbol{z}}_{k,j}$ is the j-th token of the k-th instance after standard Layer Normalization (applied to the entire packed sequence $\boldsymbol{z}$ to produce $\hat{\boldsymbol{z}}$), the AdaLN operation is:

$$\text{AdaLN}(\hat{\boldsymbol{z}}_{k,j}, \mathbf{c}_k) = \boldsymbol{\gamma}_k \odot \hat{\boldsymbol{z}}_{k,j} + \boldsymbol{\beta}_k, \tag{8}$$

where $\odot$ denotes element-wise multiplication. This ensures that adaptive normalization is applied consistently and specifically to each sub-sequence of tokens based on its original data instance, maintaining fidelity of conditioning within the computationally efficient packed representation.

[5] Following the common practice [4, 27, 47, 69, 74, 83], we use $\theta = 10000$ to ensure distinct positional signals over typical sequence lengths.

Table 1: **Benchmarking class-conditional image generation on standard *ImageNet* 256×256 and 512×512 benchmarks.** Notably, **a single NiT model can compete on both two benchmarks**. "↓" or "↑" indicate lower or higher values are better. "# Res" and "# Token" respectively represent the resolution, total training token budget. "mFID" denotes the average Fréchet inception distance (FID) value of two benchmarks. "†": an independent model is required for each benchmark, leading to a cumulative computational cost, as reflected by the huge "# Token". "*": increasing the training steps of NiT-XL. All the results are reported with the utilization of classifier-free-guidance (CFG).

Method	# Param	# Res	# Token	256×256					512×512					mFID↓
				FID↓	sFID↓	IS↑	Prec.↑	Rec.↑	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑	
Auto-regressive Models for Specific Resolutions														
MaskGiT	-	256	-	6.18	-	182.1	0.80	0.51	-	-	-	-	-	-
LlamaGen-3B	3B	384	$221B$	2.18	5.96	263.33	0.82	0.58	-	-	-	-	-	-
VAR-2B†	2B	256	-	1.73	-	**350.2**	0.82	0.60	2.63	-	303.2	-	-	2.18
Diffusion Models for Specific Resolutions														
DiT-XL/2†	675M	256&512	$1428B$	2.27	4.60	278.24	**0.83**	0.57	3.04	5.02	240.82	0.84	0.54	2.66
SiT-XL/2†	675M	256&512	$1428B$	2.06	4.50	270.3	0.82	0.57	2.62	4.18	252.2	0.84	0.57	2.34
FlowDCN†	675M	256&512	$158B$	2.00	**4.37**	263.16	0.82	0.58	2.44	4.53	252.8	0.84	0.54	2.22
FiTv2-XL†	671M	256&512	$237B$	2.26	4.44	293.83	0.80	0.62	2.62	6.63	**307.54**	0.80	0.57	2.44
SiT-REPA†	675M	256&512	$525B$	**1.42**	4.70	305.7	0.80	**0.65**	2.08	4.19	274.6	**0.83**	0.58	**1.75**
EDM2-L	777M	512	$472B$	-	-	-	-	-	1.88	4.27	258.21	0.81	0.58	-
EDM2-XXL	1.5B	512	$472B$	-	-	-	-	-	1.81	-	-	-	-	-
Generalist Diffusion Models for Arbitrary Resolution														
NiT-XL	675M	Native	$131B$	2.16	6.34	253.44	0.79	0.62	1.57	4.13	260.69	0.81	**0.63**	1.86
NiT-XL*	675M	Native	$183B$	2.06	6.34	256.21	0.80	0.62	**1.48**	**4.07**	262.23	0.81	0.62	1.77

Conclusion. NiT's architecture design of native-resolution generative modeling fundamentally enhances image synthesis. By preserving the complete spatial hierarchy and detail of original inputs, NiT intrinsically learns scale-independent visual distributions. This leads to superior fidelity and adaptability in zero-shot generalization across diverse resolutions and aspect ratios.

4 Experiments

4.1 Setup

Native-Resolution Generation Evaluation. To comprehensively evaluate the resolution generalization, we conduct a wide range of resolution spectra.

- We evaluate NiT on standard 256×256 and 512×512 benchmarks with **a single model**, differing from the previous implementations with two distinct models.
- For high-resolution generalization evaluation, experiments are conducted on four resolutions: $\{768 \times 768, 1024 \times 1024, 1536 \times 1536, 2048 \times 2048\}$.
- For aspect ratio generalization analysis, experiments are conducted on six aspect ratios: $\{1:3, 9:16, 3:4, 4:3, 16:9, 3:1\}$. The corresponding resolutions are: $\{320 \times 960, 432 \times 768, 480 \times 640, 640 \times 480, 768 \times 432, 960 \times 320\}$.

Implementation Details. We use DC-AE [12] with a $32\times$ down-sampling scale and 32 latent dimensions as our image encoder. Therefore, an image with the shape of $H \times W$ is encoded into a latent token vector $\mathbf{z} \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32} \times 32}$. For class-guided image generation experiments, the model architecture follows DiT [56], except for using a patch size of 1. Our model is trained with native-resolution images, so the batch size is unsuitable in our setting. Therefore, similar to LLMs [22, 82], we use **token budget** (*i.e.*, the summation of total tokens in all training iterations) to represent the training compute. For class-guided image generation, we use $131,072$[6] tokens in one iteration. Unless otherwise stated, all results in Tabs. 1 to 3 are evaluated with the NiT model trained for $1000K$ steps. We report FID [30], sFID [51], Inception Score (IS) [63], Precision and Recall [41] using ADM evaluation suite [18]. Text-to-image generation results are detailed in Appendix A.

[6]It holds: $131,072 = 256 \times 512 = 1024 \times 128$. For 256×256 resolution (256 tokens each in DiT [56]), this corresponds to a larger batch of 512 images. Conversely, for higher-resolution 512×512 images (1024 tokens each), it corresponds to a smaller batch of 128 images is used to maintain the same total token count.

Table 2: **Benchmarking resolution generalization capabilities on *ImageNet*.** As FiTv2 and SiT-REPA have failed to generalize to 1024×1024 resolution, we use "✗" to represent their inabilities to generalize to higher resolutions.

Method	768×768			1024×1024			1536×1536			2048×2048		
	FID↓	**sFID↓**	**IS↑**	**FID↓**	**sFID↓**	**IS↑**	**FID↓**	**sFID↓**	**IS↑**	**FID↓**	**sFID↓**	**IS↑**
EDM2-L	9.02	18.57	248.15	40.74	47.29	119.41	105.57	69.31	40.05	172.30	89.18	16.82
FlowDCN	9.817	24.52	202.86	18.64	42.36	206.66	41.170	61.75	150.22	69.88	68.15	81.33
FiTv2-XL	190.69	143.78	8.56	281.55	209.20	4.55	✗	✗	✗	✗	✗	✗
SiT-REPA	274.63	215.25	3.58	286.79	235.07	2.643	✗	✗	✗	✗	✗	✗
NiT-XL	**4.05**	**8.77**	**262.31**	**4.52**	**7.99**	**286.87**	**6.51**	**9.97**	**230.10**	**24.76**	**18.01**	**131.36**

4.2 State-of-the-Art Class-Guided Image Generation

Standard Benchmarks We first demonstrate the effectiveness of NiT on standard *ImageNet* 256×256 and 512×512 benchmarks. We compare NiT with state-of-the-art autoregressive models: MaskGit [8], LlamaGen [70], and VAR [76], as well as state-of-the-art diffusion models: DiT [49], SiT [48], FlowDCN [79], FiTv2 [80], SiT-REPA [86], and EDM2 [37]. All these are resolution-expert methods, independently training two models for the two benchmarks.

Performance Analysis. As demonstrated in Tab. 1, NiT-XL achieves **the best FID 1.48 on the** 512×512 **benchmark**, outperforming the previous SOTA EDM2-XXL with half of the model size. On the 256×256 benchmark, our model surpasses the DiT-XL and FiTv2-XL models on FID with the same model size as well as outperforms the LlamaGen-3B model with much smaller parameters. Compared with all the baseline models, our model demonstrates **significant training efficiency**, as it avoids the cumulative computes to train two distinct models. To the best of our knowledge, this is the first time **a single model** can compete on these two benchmarks simultaneously. For *mFID* metric, NiT-XL largely outperforms DiT-XL and SiT-XL with 9.17% of the training costs. And it can be comparable with SiT-REPA, with only 25% of the token budget.

Table 3: **Benchmarking aspect ratio generalization capabilities on *ImageNet*.**"†": SiT-REPA is evaluated with 160×480, 216×384, 240×320, 320×240, 384×216 and 480×160, because only the model trained on 256-resolution image data is open-sourced. For other models, the exact resolutions are: 320×960, 432×768, 480×640, 640×480, 768×432 and 960×320.

Method	1 : 3		9 : 16		3 : 4		4 : 3		16 : 9		3 : 1	
	FID↓	**IS↑**	**FID↓**	**IS↑**	**FID↓**	**IS↑**	**FID↓**	**IS↑**	**FID↓**	**IS↑**	**FID↓**	**IS↑**
EDM2-L	32.48	68.45	8.19	170.06	5.00	183.78	5.97	170.06	11.58	144.65	39.94	59.69
FlowDCN	34.97	73.30	8.25	178.16	5.311	197.01	5.74	185.67	10.31	154.92	40.721	64.11
FiTv2-XL	67.57	37.95	50.58	60.52	49.96	59.40	62.18	46.06	71.79	42.89	94.15	35.74
SiT-REPA	147.61	14.44	34.42	94.13	3.87	242.61	4.03	242.42	37.77	88.46	114.01	18.56
NiT-XL	**16.85**	**189.18**	**4.11**	**254.71**	**3.72**	**284.94**	**3.41**	**259.06**	**5.27**	**218.78**	**9.90**	**255.05**

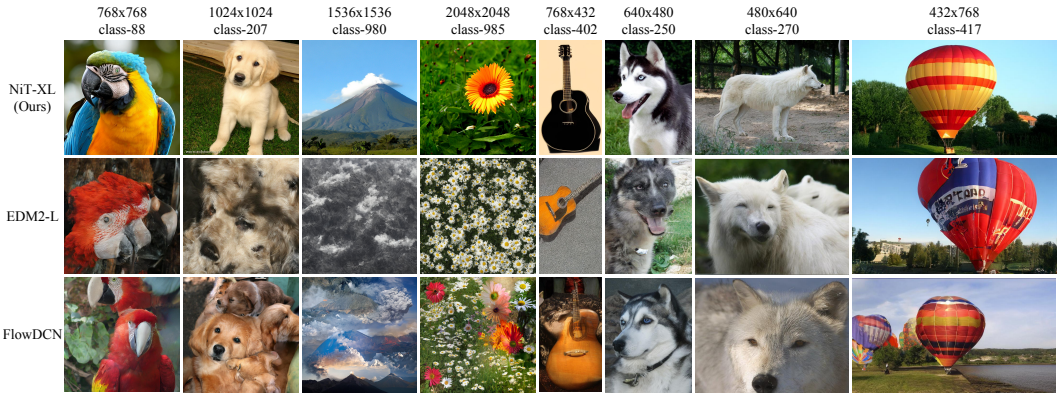


Figure 4: **Qualitative Comparison of Resolution and Aspect Ratio Generalization.**We provide the visualization of NiT, EDM2 and FlowDCN, because FiTv2 and SiT-REPA demonstrate inferior generalization capability revealed by quantitative results.

Table 4: **Ablation study on resolution generalization.** We explore different data mixtures to realize why NiT can generalize to unseen resolutions. All these models are trained for $200K$ Steps.

Data Mixture	256×256			512×512			768×768		
	FID↓	**sFID**↓	**IS**↑	**FID**↓	**sFID**↓	**IS**↑	**FID**↓	**sFID**↓	**IS**↑
(a) Native Resolution	23.22	15.61	63.50	15.75	12.44	110.50	16.77	14.74	115.54
(b) Native Resolution + 256 + 512	15.46	7.67	86.57	9.56	5.25	137.26	12.42	13.51	149.09
(c) 256 + 512	15.94	7.73	83.29	9.63	6.05	134.70	33.50	99.63	81.58

Table 5: **Ablation study on aspect ratio generalization.** We further evaluate the data mixture on different aspect ratios. All these models are trained for $500K$ Steps.

Data	1024×1024		1536×1536		2048×2048		320×960		432×768		480×640	
	FID↓	**IS**↑	**FID**↓	**IS**↑	**FID**↓	**IS**↑	**FID**↓	**IS**↑	**FID**↓	**IS**↑	**FID**↓	**IS**↑
(a)	13.15	159.52	18.11	132.53	41.95	71.19	40.14	53.41	13.45	137.31	10.63	160.91
(b)	9.73	180.29	14.29	152.37	37.79	78.66	43.86	50.25	12.14	131.96	8.55	169.24

Beyond standard benchmarks on 256×256 and 512 resolutions, we comprehensively conduct high-resolution generalization in Tab. 2 and aspect ratio generalization evaluations in Tab. 3. We compare our method with 4 widely-recognized baselines: 1) EDM2 [37], SOTA CNN-based diffusion model; 2) FlowDCN [79], CNN-based diffusion model designed for multi-resolution image generation; 3) FiTv2 [80], diffusion transformer model designed for multi-resolution image generation; 4) SiT-REPA [86], current SOTA diffusion transformer model.

Generalization Analysis As demonstrated in Tab. 2, NiT-XL significantly surpasses all the baselines on resolution generalization. Remarkably, NiT-XL achieves FID 4.07, 4.52, and 6.51 on 768, 1024×1024, and 1536×1536, respectively, demonstrating almost no performance degradation when scaling to unseen higher resolutions. EDM2-L and FlowDCN can generalize to 768×768 resolution, but they fail to generalize to higher resolutions beyond 1024×1024 resolution. FiTv2 and SiT-REPA demonstrate very inferior resolution generalization capability beyond their training resolutions. As shown in Tab. 3, NiT-XL can also generalize to arbitrary aspect ratios, greatly outperforming all the baselines. Although EDM2-L, SiT-REPA and FlowDCN can generalize to $4:3$ and $3:4$ aspect ratios, they fail to generalize more extreme aspect ratios, like $9:16$ and $1:3$. NiT-XL can generalize up to $9:16$ and $16:9$ aspect ratios with negligible performance loss, and perform best on $1:3$ and $3:1$ aspect ratios. These results indicate **NiT is initially equipped with the resolution-free generation capability**, bridging the gap between vision generation and the sequence-free generation in LLMs [1, 50, 74, 82].

The qualitative comparison in Fig. 4 is consistent with the aforementioned quantitative results. NiT demonstrates superior generalization quality than EDM2-L and FlowDCN, producing reasonable generated samples. When beyond 768 resolution, EDM2-L in particular generates images dominated by non-informative textures, while FlowDCN-XL tends to replicate objects multiple times in a single image. Regarding aspect ratio generalization, models like DM2-L and FlowDCN-XL exhibit a distinct cropping-induced bias. This strongly suggests the models have internalized a truncation bias due to their training predominantly on square or tightly cropped image samples. Consequently, they tend to generate unnaturally framed outputs with truncated object boundaries, especially when encountering extreme aspect ratios. This aligns with findings from SDXL [58], revealing that image cropping during data preparation can propagate biases into generated samples, leading to adverse effects, particularly with extreme aspect ratios. More visualizations are provided in Appendix D

4.3 Ablation Study

Set up. We find the surprising generalization ability of NiT in unseen resolutions and aspect ratios. In this part, we explore what enables the impressive generalization ability. We conduct 3 groups for ablation: (a): we only use native-resolution image data for training; (b): besides native-resolution, we add 256×256-resolution, and 512×512-resolution image data for training; 3) without native-resolution images, we only use 256×256 and 512×512 images. All the experiments are conducted using a NiT-B model with $131M$ parameters and all the results are evaluated with the usage of CFG.

As demonstrated in Tab. 4, (b) and (c) consistently beat (a) on 256×256 and 512×512 benchmarks. This is because we have not optimized the model for the two resolutions in (a), thus demonstrating

inferior performance. The performance on 256×256 and 512×512 of (b) and (c) is comparable; however, (c) extremely lags (b) on 768×768 resolution generation. We think that the training of (c) is solely on two resolutions, significantly restricting the generalization capabilities of models. Meanwhile, in Tab. 5, as we scale up training steps to 500k, we further compare the performance of (a) and (b) on resolution generalization and aspect ratio generalization. We find that (b) demonstrates stronger generalization capability than (a).

Insights. The ablation study reveals that NiT's strong generalization to unseen resolutions and aspect ratios is primarily enabled by training in native-resolution to learn a resolution- and aspect-ratio invariant visual distribution. While adding fixed resolutions like 256×256 and 512×512 improves performance on those specific sizes and aspect ratio of $1:1, 768 \times 768$, omitting native-resolution data severely hinders generalization to other aspect ratios. Therefore, a combination of varied resolution and aspect ratios, with native resolution playing a key role, is essential for robust generalization, rather than just training on a limited set of fixed scales.

Efficiency Analysis. We compare the training and inference efficiency on the ImageNet-256 benchmark using a single NVIDIA A100 GPU, revealing that NiT demonstrates better training and inference efficiency compared to DiT. Analysis is conducted with NiT-B and DiT-B model, both with $131M$ parameters. We set token number in one iteration as 65536. Specifically, NiT achieves a faster training speed of 1.28 iterations/second (iter/s) compared to DiT's 1.08 iter/s. Furthermore, NiT exhibits lower inference latency at 0.246 seconds, while DiT has a latency of 0.322 seconds, highlighting NiT's greater computational efficiency.

5 Conclusion & Limitation

In conclusion, this work introduces native-resolution image synthesis paradigm for visual content generation. We reformulate the resolution modeling as "native-resolution generation" and propose Native-resolution diffusion Transformer (NiT). To the best of our knowledge, **NiT is the first model that can achieve dual SOTA performance on 256×256 and 512×512 benchmarks in *ImageNet* generation with a single model.** We demonstrate NiT's robust generalization capabilities to unseen image resolutions and aspect ratios, significantly outperforming previous methods. While NiT shows strong performance, its generalization ability on extremely high-resolution and aspect ratios is still not satisfactory. Future research could explore optimal strategies for balancing diverse training resolutions for improved efficiency and further investigate the model's generalization limits across a wider spectrum of data domains (*e.g.*, video generation) and highly disparate aspect ratios. Additionally, the computational resources required for comprehensive multi-scale training remain a practical consideration for broader application.

A Text-to-Image Generation

A.1 Streamlined NiT Architecture for Text-to-Image Generation

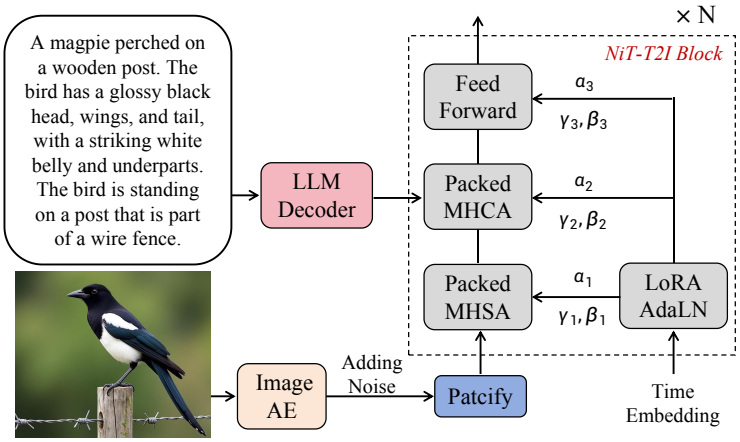


Figure 5: Illustration of NiT blocks used for text-to-image generation.

We introduce a streamlined architecture to incorporate textual information. As in Fig. 5, we insert a cross-attention block between the self-attention block and the feed-forward network. As the adaptive LayerNorm (AdaLN) module only conditions on the time embedding, we thus reduce its parameters through a LoRA [32] design. Given transformer hidden size d, the AdaLN layer predicts a tuple of all scale and shift parameters $S = [\beta_1, \beta_2, \beta_3, \gamma_1, \gamma_2, \gamma_3, \alpha_1, \alpha_2, \alpha_3]$, where β, γ, α represents the shift and scale parameters for a block, and the subscript $1, 2, 3$ denote self-attention, cross-attention, and feed-forward respectively. Based on the embedding for time step $\mathbf{t} \in \mathbb{R}^d$, S^l for the l-th NiT-T2I block is computed as:

$$S^l = [\beta_1^l, \beta_2^l, \beta_3^l, \gamma_1^l, \gamma_2^l, \gamma_3^l, \alpha_1^l, \alpha_2^l, \alpha_3^l] = W_2^l W_1^l \mathbf{t} \in \mathbb{R}^{9\times d}, \tag{9}$$

where $W_2^l \in \mathbb{R}^{(9\times d)\times r}, W_1^l \in \mathbb{R}^{r\times d}$, and the bias parameters are omitted for simplicity. We can adjust the LoRA rank r to align the block parameters with the NiT-C2I blocks.

Table 6: **Text-to-image generation of NiT**. We compare our models by the zero-shot generation task on the *COCO*-2014 [42] dataset.

Method	# Param.	Res.	FID (↓)	CLIP (↑)
DALL·E	12B	256	27.5	-
CogView2	6B	256	24.0	-
Parti-750M	750M	256	10.71	-
Parti-3B	3B	256	8.10	-
Parti-20B	20B	256	**7.23**	-
Make-A-Scene	-	256	11.84	-
Muse	3B	256	7.88	0.32
GLIDE	5B	256	12.24	-
DALL·E 2	5.5B	256	10.39	-
LDM	1.45B	256	12.63	-
Imagen	3B	256	7.27	-
eDiff-I	9B	256	6.95	-
SD-v1.5	0.9B	512	9.78	0.318
SDXL-Turbo	3B	1024	23.19	0.334
NiT-T2I	673M	1024	9.18	**0.345**

A.2 Advanced Text-to-Image Generation

Implementation details. For text-to-image generation, and adopt Gemma3-1B-it [73] as our text encoder. We use LoRA rank $r = 192$ in AdaLN, which leads to a total $673M$ model parameters, matching the model parameters of the XL model in the C2I (class-to-image) setting. We use the REPA [86] strategy in the training, where a RADIO-v2.5-H [29] model serves as the visual encoder. We conduct text-to-image generation experiments on the SAM [39] dataset with captions generated by MiniCPM-V [84]. We use a token number in each iteration as $786,432$ and train the model for $400K$ steps and evaluate image quality using COCO-val-2014 [42] benchmark.

Compared baselines and Results. We report the zero-shot text-to-image performance on *COCO*-2014 benchmark in Tab. 6, competing with DALL·E [61], CogView2 [19], Parti [85], Make-A-Scene [24], Muse [7], GLIDE [52], DALL·E 2 [60], LDM [62], eDiff-I [5], SD-v1.5 [62], and SDXL-Turbo [64]. NiT-T2I demonstrates competitive performance with the baseline models. Notably, NiT-T2I achieves the best CLIP score of 0.345, and it surpasses SD-v1.5 and SDXL-Turbo on both FID and CLIP scores with a much smaller model size and training costs.

B Detailed Quantitative Results

In this section, we report all the metrics of generalization experiments in Tab. 7. In addition, we provide the CFG (classifier-free-guidance) hyperparameters of NiT-XL.

Table 7: **Detailed Quantitative Results of NiT-XL.** We further provide the CFG scale and interval for each experiment and report all the metric values for generalization experiments.

Resolution	CFG-scale	CFG-interval	**FID↓**	**sFID↓**	**IS↑**	**Prec.↑**	**Rec.↑**
256×256	2.25	[0.0, 0.7]	2.16	6.34	253.44	0.79	0.62
512×512	2.05	[0.0, 0.7]	1.57	4.13	260.69	0.81	0.63
768×768	3.0	[0.0, 0.7]	4.05	8.77	262.31	0.83	0.52
1024×1024	3.0	[0.0, 0.8]	4.52	7.99	286.87	0.82	0.50
1536×1536	1.5	[0.0, 1.0]	6.51	9.97	230.10	0.83	0.42
2048×2048	1.5	[0.0, 1.0]	24.76	18.02	131.36	0.67	0.46
320×960	4.0	[0.0, 0.9]	16.85	17.79	189.18	0.71	0.38
432×768	2.75	[0.0, 0.7]	4.11	10.30	254.71	0.83	0.55
480×640	2.75	[0.0, 0.7]	3.72	8.23	284.94	0.83	0.54
640×480	2.5	[0.0, 0.7]	3.41	8.07	259.06	0.83	0.56
768×432	2.85	[0.0, 0.7]	5.27	9.92	218.78	0.80	0.55
960×320	4.5	[0.0, 0.9]	9.90	25.78	255.95	0.74	0.40

C Detailed Implementation of NiT

This section demonstrates the detailed implementation of Packed Full-Attention and Packed Adaptive Layer Normalization (AdaLN) in an NiT block. Different from traditional attention implementation for batched data, we use FlashAttention2 [15] for packed data, leading to enhanced efficiency, as in Algorithm 1. Besides, we use a broadcast mechanism on conditional vector $\mathbf{c}$ for Packed-AdaLN, detailed in Algorithm 2.

D Qualitative Results of NiT

D.1 Generalization Comparison with Baseline Models

We provide the qualitative results of NiT-XL, EDM2-L [37], and FlowDCN-XL [79] on resolution generalization and aspect ratio generalization. The qualitative results of FiTv2-XL [80] and REPA-XL [86] are not provided because these two methods demonstrate very weak generalization capabilities. The resolution generalization visualizations are shown in Figs. 6 to 9, while the aspect ratio generalization visualizations are demonstrated in Figs. 10 to 15.

Algorithm 1 Packed Full-Attention with FlashAttention for flexible-length sequence processing.

```python

import torch
import torch.nn as nn
from flash_attn import flash_attn_varlen_func

def rotate_half(x):
    x = rearrange(x, '... (d r) -> ... d r', r = 2)
    x1, x2 = x.unbind(dim = -1)
    x = torch.stack((-x2, x1), dim = -1)
    return rearrange(x, '... d r -> ... (d r)')

class Attention(nn.Module):
    def __init__(self,
        dim: int,
        num_heads: int = 8,
        qkv_bias: bool = False,
        qk_norm: bool = False,
        attn_drop: float = 0.,
        proj_drop: float = 0.,
        norm_layer: nn.Module = nn.LayerNorm,
    ) -> None:
        super().__init__()
        assert dim % num_heads == 0, "dim should be divisible by num_heads"
        self.num_heads = num_heads
        self.head_dim = dim // num_heads
        self.scale = self.head_dim ** -0.5
        self.qkv = nn.Linear(dim, dim * 3, bias=qkv_bias)
        self.q_norm = norm_layer(self.head_dim) if qk_norm else nn.Identity()
        self.k_norm = norm_layer(self.head_dim) if qk_norm else nn.Identity()
        self.attn_drop = nn.Dropout(attn_drop)
        self.proj = nn.Linear(dim, dim)
        self.proj_drop = nn.Dropout(proj_drop)

    def forward(self,
        x: torch.Tensor,
        cu_seqlens: torch.Tensor,
        freqs_cos: torch.Tensor,
        freqs_sin: torch.Tensor
    ) -> torch.Tensor:
        # x: packed sequence with shape [N, D]
        # cu_seqlens: [0, h_1*w1, h_1*w_1+h_2*w_2, ...], the cumulated sequence length
        # freqs_cos, freqs_sin: 2D-RoPE frequences
        N, C = x.shape
        qkv = self.qkv(x).reshape(
            N, 3, self.num_heads, self.head_dim
        ).permute(1, 0, 2, 3)
        ori_dtype = qkv.dtype
        q, k, v = qkv.unbind(0)
        q, k = self.q_norm(q), self.k_norm(k)

        # Use axial 2D-RoPE to inject 2D structural priors
        q = q * freqs_cos + rotate_half(q) * freqs_sin
        k = k * freqs_cos + rotate_half(k) * freqs_sin
        q, k = q.to(ori_dtype), k.to(ori_dtype)

        max_seqlen = (cu_seqlens[1:] - cu_seqlens[:-1]).max().item()

        # apply flash-attn for efficient implementation
        x = flash_attn_varlen_func(
            q, k, v, cu_seqlens, cu_seqlens, max_seqlen, max_seqlen
        ).reshape(N, -1)

        x = self.proj(x)
        x = self.proj_drop(x)
        return x
```

Algorithm 2 Packed Adaptive Layer Normalization and NiT block.

```python

import torch
import torch.nn as nn
from timm.models.vision_transformer import Mlp


def modulate(x, shift, scale):
    return x * (1 + scale) + shift

class NiTBlock(nn.Module):
    """
    A NiT block with adaptive layer norm zero (adaLN-Zero) conditioning.
    """
    def __init__(self, hidden_size, num_heads, mlp_ratio=4.0, **block_kwargs):
        super().__init__()
        self.norm1 = nn.LayerNorm(hidden_size, elementwise_affine=False, eps=1e-6)
        self.attn = Attention(
            hidden_size, num_heads=num_heads, qkv_bias=True,
            qk_norm=block_kwargs["qk_norm"]
        )
        self.norm2 = nn.LayerNorm(hidden_size, elementwise_affine=False, eps=1e-6)
        mlp_hidden_dim = int(hidden_size * mlp_ratio)
        approx_gelu = lambda: nn.GELU(approximate="tanh")
        self.mlp = Mlp(
            in_features=hidden_size, hidden_features=mlp_hidden_dim,
            act_layer=approx_gelu, drop=0
        )
        self.adaLN_modulation = nn.Sequential(
            nn.SiLU(),
            nn.Linear(hidden_size, 6 * hidden_size, bias=True)
        )

    def forward(self, x, c, hw_list, freqs_cos, freqs_sin):
        # x: packed sequence with shape [N, D]
        # c: conditional vector with shape [n, D], (n represetns number of instances)
        # hw_list: [[h1_, w_1], [h2, w_2], ..., [h_n, w_n]]
        # freqs_cos, freqs_sin: 2D-RoPE frequences

        seqlens = hw_list[:, 0] * hw_list[:, 1]
        cu_seqlens = torch.cat([
            torch.tensor([0], device=hw_list.device, dtype=torch.int),
            torch.cumsum(seqlens, dim=0, dtype=torch.int)
        ])
        # (n, D) -> (N, D) for Packed-AdaLN
        c = torch.cat([c[i].unsqueeze(0).repeat(seqlens[i], 1) for i in range(B)], dim=0)

        # predict all the shift-and-scale parameters with _acked-AdaLN
        (
            shift_msa, scale_msa, gate_msa, shift_mlp, scale_mlp, gate_mlp
        ) = self.adaLN_modulation(c).chunk(6, dim=-1)

        x = x + gate_msa * self.attn(
            modulate(self.norm1(x), shift_msa, scale_msa),
            cu_seqlens, freqs_cos, freqs_sin
        )
        x = x + gate_mlp * self.mlp(modulate(self.norm2(x), shift_mlp, scale_mlp))

        return x
```

Based on these visualization results, we find that NiT-XL achieves the best qualitative performance on both resolution generalization and aspect ratio generalization, **consistent with its best FIDs and IS scores** in the manuscript. We then provide more analysis on the qualitative results.

Analysis of resolution generalization visualization. As demonstrated in Figs. 6 to 9, NiT-XL demonstrates almost no quality degradation from 768×768 to 1536×1536 resolution. It can also generate reasonable content on 2048×2048 resolution. However, EDM2-L and FlowDCN-XL demonstrate inferior visual quality in resolution generalization. Although EDM2-L and FlowDCN-XL can generate some plausible samples on 768×768 resolution, they fail to generalize to higher resolutions (1024×1024 to 2048×2048). The key limitations are:

1. *Lack of Semantic Coherence.* Both EDM2-L and FlowDCN-XL frequently fail to generate identifiable, realistic instances of the target classes, especially for high resolution (see Fig. 9).
2. *Repetitive Textures.* EDM2-L in particular generates images dominated by repetitive, non-informative textures, lacking structural variation or clear object boundaries.
3. *Object Duplication and Spatial Disruption.* FlowDCN-XL tends to replicate objects multiple times in a single image, resulting in cluttered and spatially implausible compositions.
4. *Color and Lighting Artifacts.* EDM2-L often outputs grayscale or dull images, while FlowDCN-XL introduces unnatural color schemes and poor lighting consistency.

These limitations reveal that neither model effectively integrates objects into realistic backgrounds, with EDM2-L omitting context and FlowDCN-XL producing jumbled scenes. There is **an evident reliance on local textures at the expense of global object structure and semantics**.

Analysis of aspect ratio generalization visualization. As demonstrated in Figs. 10 to 15, NiT-XL demonstrate superior aspect ratio generalization performance than EDM2-L and FlowDCN-XL. Compared to the NiT-XL, both EDM2-L and FlowDCN-XL exhibit several notable shortcomings in image generation quality:

1. *Cropping-Induced Bias.* Long or wide objects, such as guitars (i.e., *class-402*) or parrots (i.e., *class-88,89*), are often truncated or improperly framed. This suggests **a form of information leakage or truncation bias introduced by over-reliance on square or tightly cropped training samples**, leading to unnaturally framed and truncated object boundaries. This corresponds to the finding in SDXL [58]: cropping image data can leak into the generated samples, causing malicious effects, especially in extreme aspect ratios (see Figs. 10 and 11).
2. *Blurring and Lack of Detail.* Many generated images, such as the sea turtle (*class-33*) in FlowDCN-XL or the flowers (*class-985*) in EDM2-L, lack the sharpness and textural richness seen in the NiT-XL outputs. This indicates poor high-frequency detail modeling, which compromises the realism and clarity of the outputs.
3. *Object Repetition and Spatial Artifacts.* There is frequent duplication of objects and structurally incoherent arrangements, breaking spatial consistency. Some entities appear anatomically incorrect or exhibit unnatural part arrangements, especially in animal classes.
4. *Color Artifacts.* Inconsistent coloring and unnatural saturation further diminish the realism of the generated images.

The visualization highlights **a critical limitation of conventional training pipelines that rely on center cropping and resizing to square resolutions**. Models like EDM2-L and FlowDCN-XL, which were trained under such regimes, often fail to generalize to objects with other aspect ratios. This is particularly evident in examples such as guitars (*class-402*), parrots (*class-88,89*), and volcanoes (*class-980*) that are naturally elongated either horizontally or vertically. In contrast, NiT-XL demonstrates robust aspect ratio generalization, preserving the spatial integrity and composition of elongated objects without distortion, demonstrating the effectiveness of its native resolution modeling.

D.2 More Qualitative Results of NiT

More qualititative results of NiT-XL are demonstrated in Figs. 16 to 25

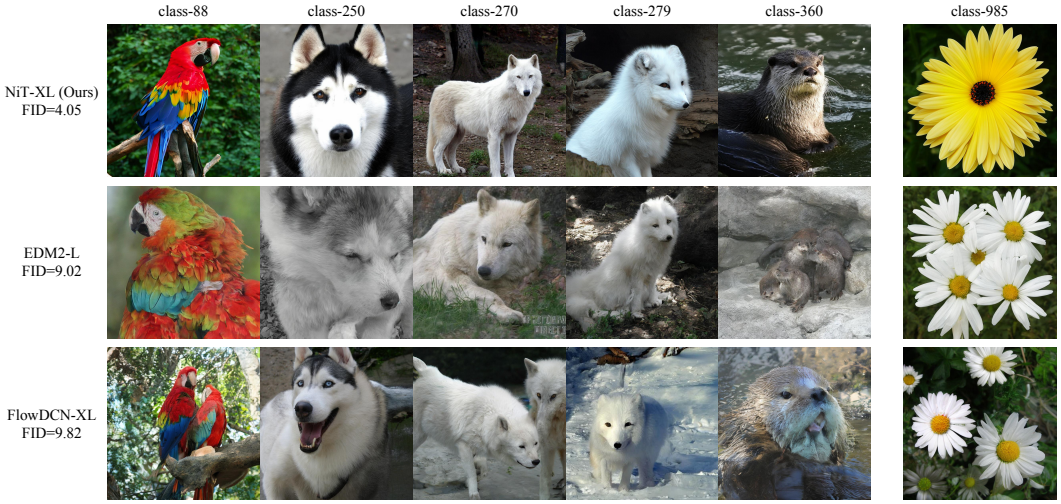


Figure 6: Qualitative comparison of resolution generalization on 768×768 resolution.

Figure 7: Qualitative comparison of resolution generalization on 1024×1024 resolution.

Figure 8: Qualitative comparison of resolution generalization on 1536×1536 resolution.

Figure 9: Qualitative comparison of resolution generalization on 2048×2048 resolution.

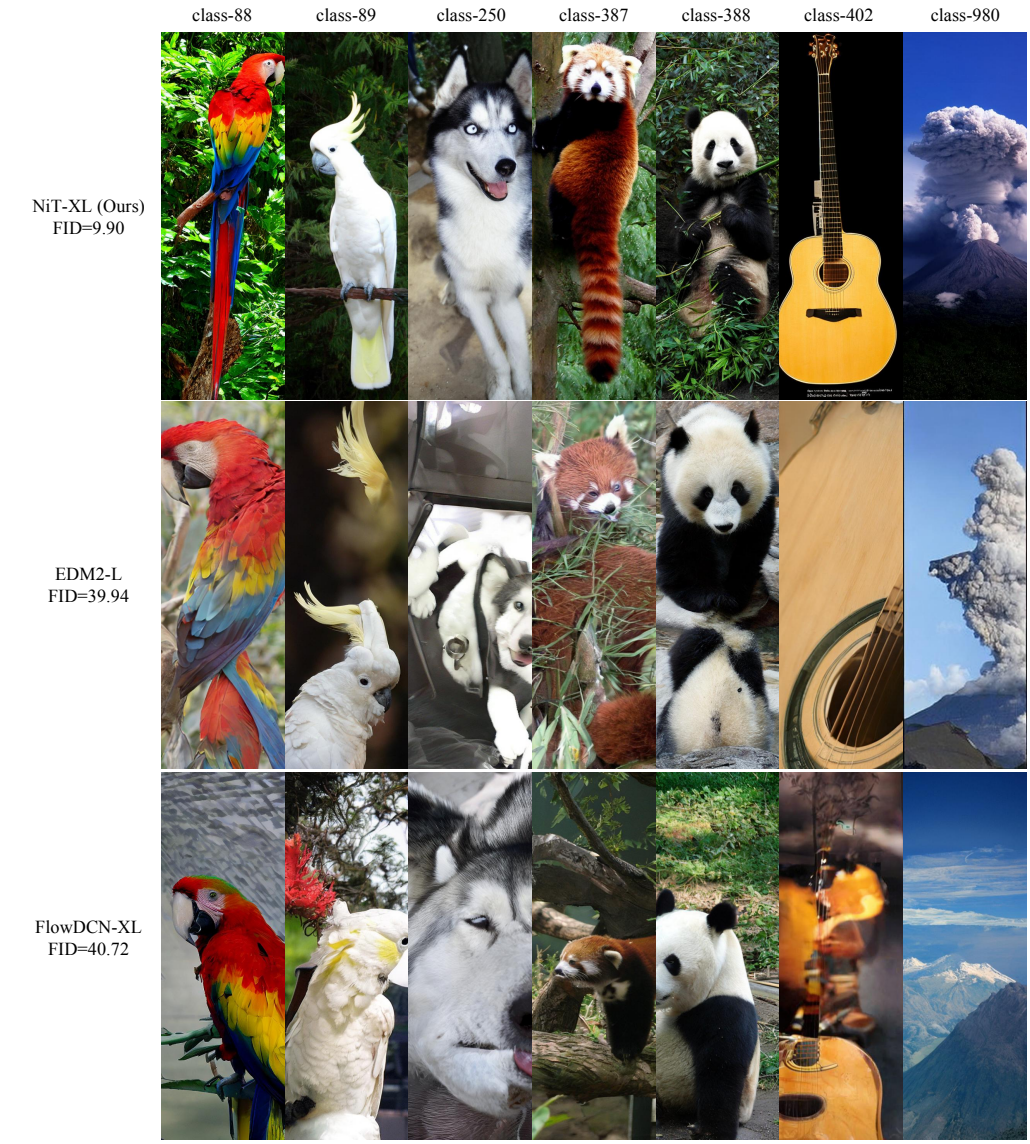


Figure 10: Qualitative comparison of aspect ratio generalization on 960×320 resolution (corresponding to $3:1$ aspect ratio).

Figure 11: Qualitative comparison of aspect ratio generalization on 320×960 resolution (corresponding to $1:3$ aspect ratio).

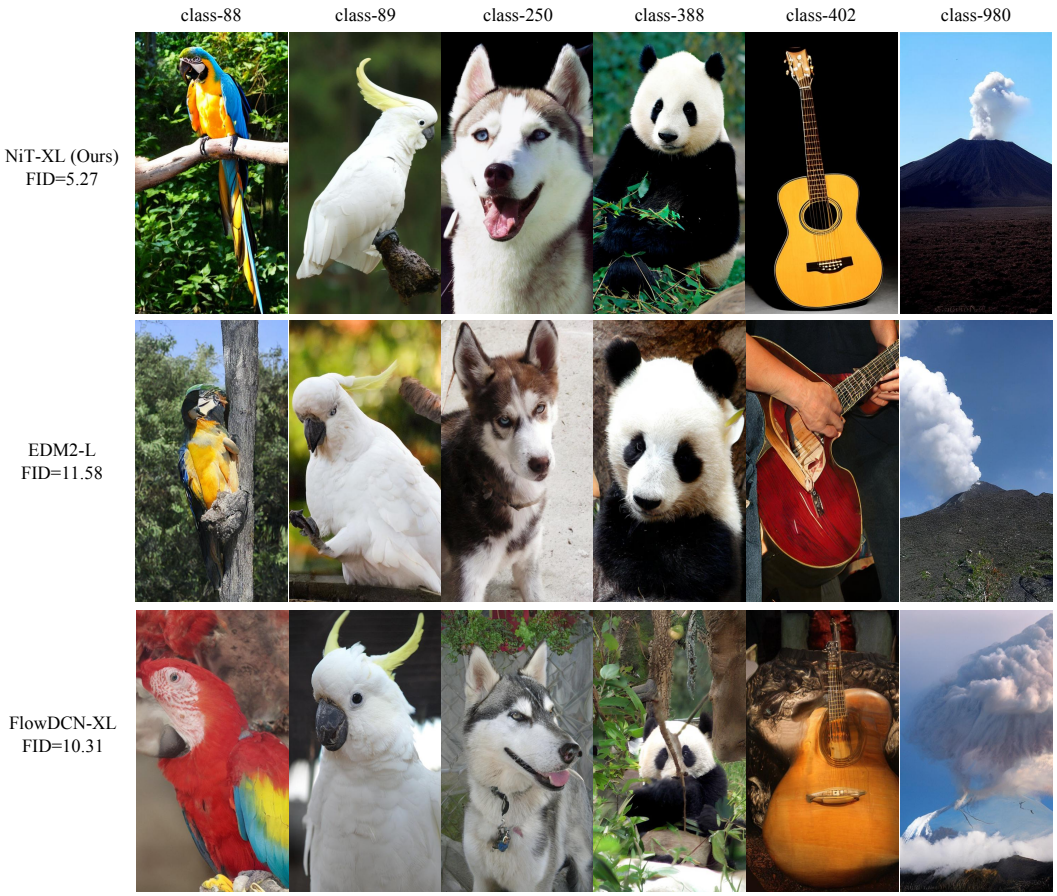


Figure 12: Qualitative comparison of aspect ratio generalization on 768×432 resolution (corresponding to $16:9$ aspect ratio).

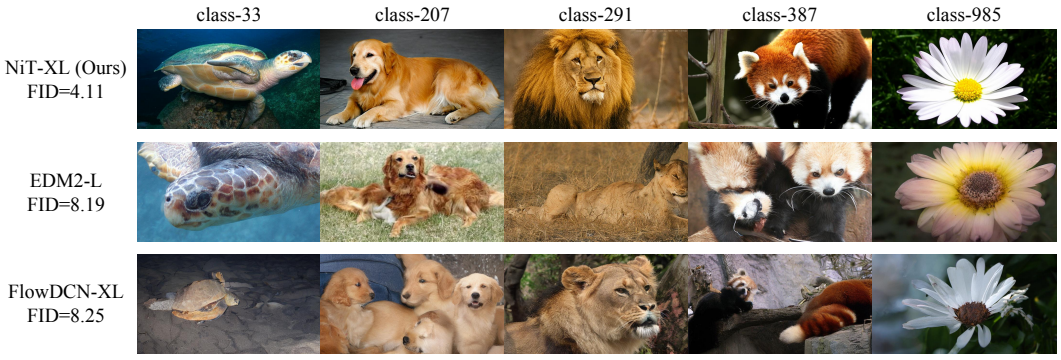


Figure 13: Qualitative comparison of aspect ratio generalization on 432×768 resolution (corresponding to $9:16$ aspect ratio).

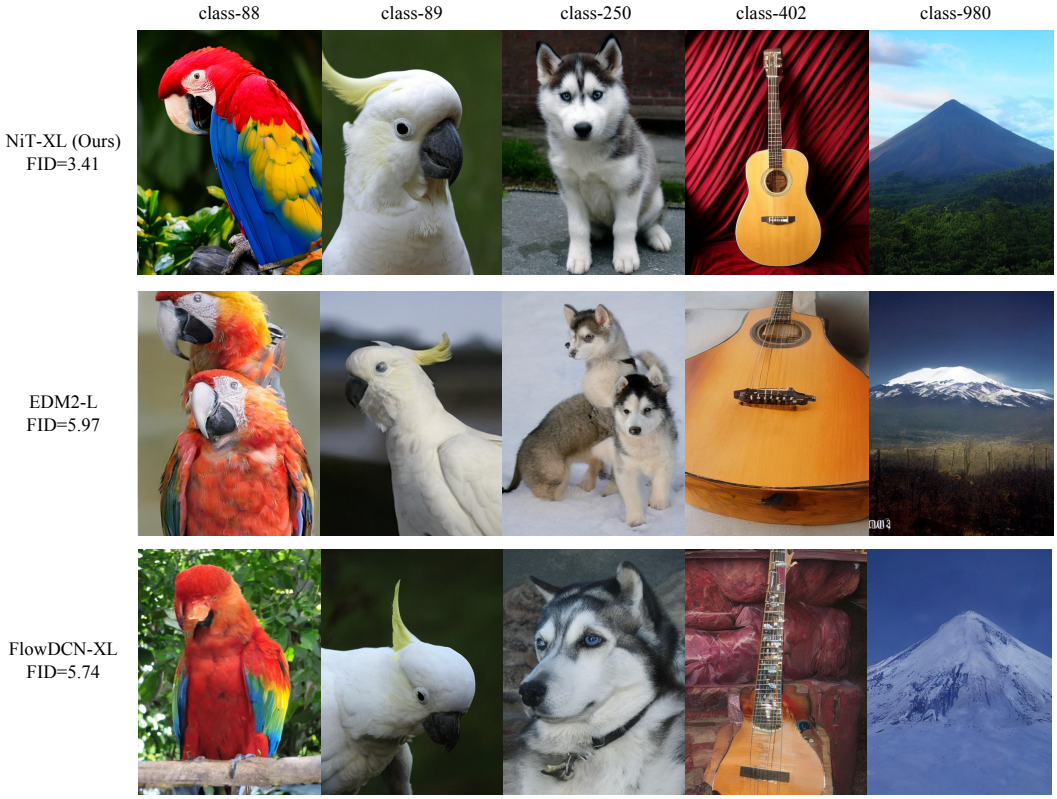


Figure 14: Qualitative comparison of aspect ratio generalization on 640×480 resolution (corresponding to $4:3$ aspect ratio).

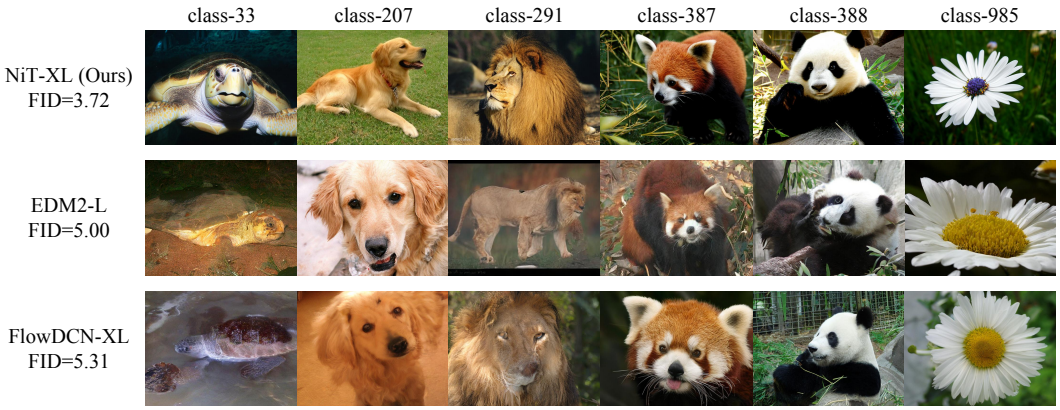


Figure 15: Qualitative comparison of aspect ratio generalization on 480×640 resolution (corresponding to $3:4$ aspect ratio).

Figure 16: Uncurated generation results of NiT-XL. We use the class label as 33.

Figure 17: Uncurated generation results of NiT-XL. We use the class label as 88.

Figure 18: Uncurated generation results of NiT-XL. We use the class label as 250.

Figure 19: Uncurated generation results of NiT-XL. We use the class label as 279.

Figure 20: Uncurated generation results of NiT-XL. We use the class label as 388.

Figure 21: Uncurated generation results of NiT-XL. We use the class label as 417.

Figure 22: Uncurated generation results of NiT-XL. We use the class label as 437.

Figure 23: Uncurated generation results of NiT-XL. We use the class label as 812.

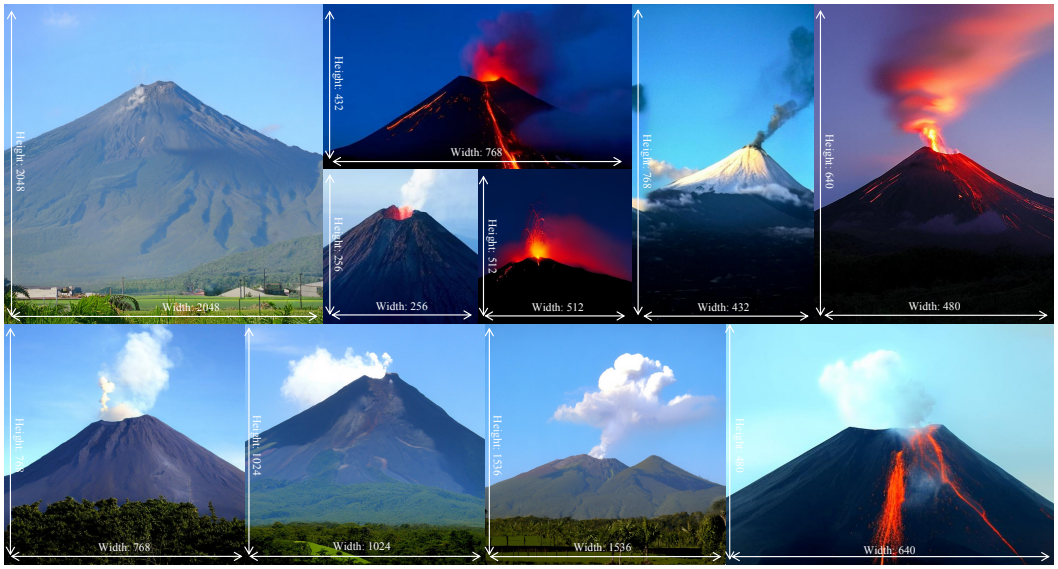


Figure 24: Uncurated generation results of NiT-XL. We use the class label as 980.

Figure 25: Uncurated generation results of NiT-XL. We use the class label as 985.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

[3] Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

[4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[5] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, et al. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 2020.

[7] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023.

[8] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[9] Guanzheng Chen, Xin Li, Zaiqiao Meng, Shangsong Liang, and Lidong Bing. Clex: Continuous length extrapolation for large language models, 2024. *URL https://arxiv. org/abs/2310.16450*.

[10] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-sigma: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. In *European Conference on Computer Vision*, pages 74–91. Springer, 2024.

[11] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-alpha: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023.

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

[13] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

[14] Ta-Chung Chi, Ting-Han Fan, Peter J. Ramadge, and Alexander Rudnicky. Kerple: Kernelized relative positional embedding for length extrapolation. In *NeurIPS*, 2022.

[15] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

[16] Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Patch n'pack: Navit, a vision transformer for any aspect ratio and resolution. *arXiv preprint arXiv:2307.06304*, 2023.

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[18] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 2021.

[19] Ming Ding, Wendi Zheng, Wenyi Hong, and Jie Tang. Cogview2: Faster and better text-to-image generation via hierarchical transformers. *Advances in Neural Information Processing Systems*, 35:16890–16902, 2022.

[20] Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. Longrope: Extending llm context window beyond 2 million tokens. *arXiv preprint arXiv:2402.13753*, 2024.

[21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[22] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[23] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. 2024.

[24] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *European Conference on Computer Vision*, pages 89–106. Springer, 2022.

[25] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer. *arXiv preprint arXiv:2303.14389*, 2023.

[26] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[27] Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, et al. Seed1. 5-vl technical report. *arXiv preprint arXiv:2505.07062*, 2025.

[28] Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. Lm-infinite: Zero-shot extreme length generalization for large language models. *arXiv preprint arXiv:2308.16137*, 2023.

[29] Greg Heinrich, Mike Ranzinger, Yin Hongxu, Yao Lu, Jan Kautz, Andrew Tao, Bryan Catanzaro, and Pavlo Molchanov. Radiov2. 5: Improved baselines for agglomerative vision foundation models. In *Proc. CVPR*, volume 2, page 6, 2025.

[30] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 2017.

[31] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020.

[32] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.

[33] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. 2005.

[34] Bernd Jähne. *Digital image processing*. Springer Science & Business Media, 2005.

[35] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[36] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *NeurIPS*, 2022.

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

[38] Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. The impact of positional encoding on length generalization in transformers. *Advances in Neural Information Processing Systems*, 36:24892–24928, 2023.

[39] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[40] Mario Michael Krell, Matej Kosec, Sergio P Perez, and Andrew Fitzgibbon. Efficient sequence packing without cross-contamination: Accelerating large language models without impacting performance. *arXiv preprint arXiv:2107.02027*, 2021.

[41] T. Kynkäänniemi, T. Karras, S. Laine, and T Lehtinen, J.and Aila. Improved precision and recall metric for assessing generative models. *NeurIPS*, 2019.

[42] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[43] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[44] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[45] Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

[46] 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.

[47] Zeyu Lu, ZiDong Wang, Di Huang, Chengyue Wu, Xihui Liu, Wanli Ouyang, and LEI BAI. Fit: Flexible vision transformer for diffusion model. 2024.

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

[49] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

[50] AI Meta. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation, april 2025, 2025.

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

[52] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[53] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.

[54] Alan V Oppenheim, Alan S Willsky, and Syed Hamid Nawab. *Signals & systems*. Pearson Educación, 1997.

[55] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[56] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

[57] Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*, 2023.

[58] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[59] Ofir Press, Smith Noah, and Lewis Mike. Train short, test long: Attention with linear biases enables input length extrapolation. In *ICLR*, 2021.

[60] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[61] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021.

[62] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[63] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X Chen. Improved techniques for training gans. *NeurIPS*, 2016.

[64] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pages 87–103. Springer, 2024.

[65] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *NeurIPS*, 2019.

[66] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *NeurIPS*, 2020.

[67] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[68] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis and machine vision*. Springer, 2013.

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

[70] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

[71] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[72] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[73] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

[74] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

[75] Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, et al. Kimi-vl technical report. *arXiv preprint arXiv:2504.07491*, 2025.

[76] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

[77] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, and Baptiste Rozière et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

[78] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, and Nikolay Bashlykov et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

[79] Shuai Wang, Zexian Li, Tianhui Song, Xubin Li, Tiezheng Ge, Bo Zheng, and Limin Wang. Flowdcn: Exploring dcn-like architectures for fast image generation with arbitrary resolution. *arXiv preprint arXiv:2410.22655*, 2024.

[80] ZiDong Wang, Zeyu Lu, Di Huang, Cai Zhou, Wanli Ouyang, and LEI BAI. Fitv2: Scalable and improved flexible vision transformer for diffusion model. *arXiv preprint arXiv:2410.13925*, 2024.

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

[82] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[83] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

[84] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024.

[85] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022.

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

[87] Dawei Zhu, Nan Yang, Liang Wang, Yifan Song, Wenhao Wu, Furu Wei, and Sujian Li. Pose: Efficient context window extension of llms via positional skip-wise training. *arXiv preprint arXiv:2309.10400*, 2023.