FlexVAR: Flexible Visual Autoregressive Modeling without Residual Prediction

Siyu Jiao[1] **Gengwei Zhang**[2] **Yinlong Qian**[3] **Jiancheng Huang**[3] **Yao Zhao**[1]
Humphrey Shi[4] **Lin Ma**[3] **Yunchao Wei**[1†] **Zequn Jie**[3†]

[1] Institute of Information Science, Beijing Jiaotong University
[2] University of Technology Sydney
[3] Meituan
[4] Georgia Institute of Technology

Code: FlexVAR

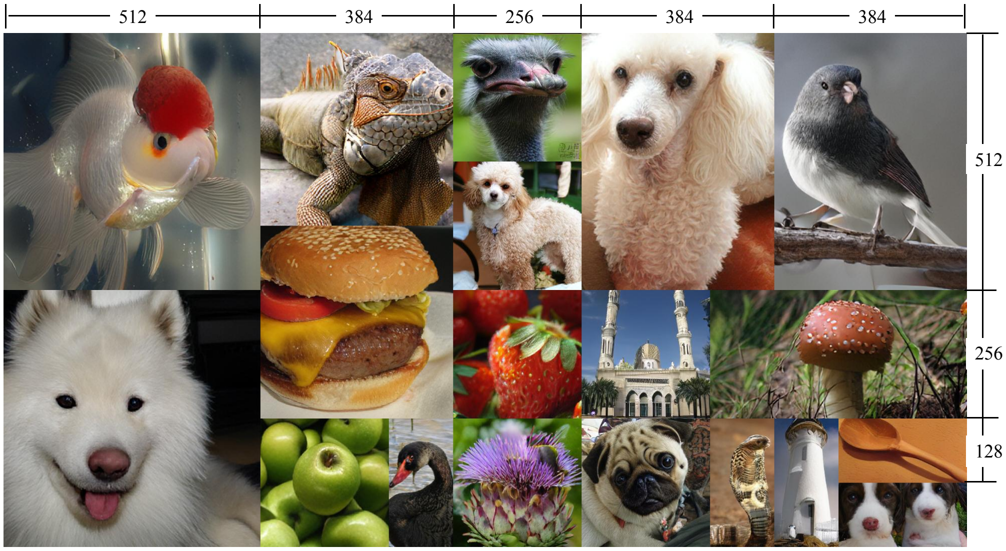

Figure 1: Generated samples from FlexVAR-$d24$ (1.0B). FlexVAR generates images with various resolutions and aspect ratios, even though it is trained with a resolution of $\leq 256\times256$.

Abstract

This work challenges the residual prediction paradigm in visual autoregressive modeling and presents FlexVAR, a new Flexible Visual AutoRegressive image generation paradigm. FlexVAR facilitates autoregressive learning with ground-truth prediction, enabling each step to independently produce plausible images. This simple, intuitive approach swiftly learns visual distributions and makes the generation process more flexible and adaptable. **Trained solely on low-resolution images** ($\leq$ 256px), FlexVAR can: (1) Generate images of various resolutions and aspect ratios, even exceeding the resolution of the training images. (2) Support various image-to-image tasks, including image refinement, in/out-painting, and image expansion. (3) Adapt to various autoregressive steps, allowing for faster inference with fewer steps or enhancing image quality with more steps. Our 1.0B model outperforms its VAR counterpart on the ImageNet 256×256 benchmark. Moreover, when zero-shot transfer the image generation process with 13 steps, the performance further improves to 2.08 FID, outperforming state-of-the-art

[†] Corresponding author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

autoregressive models AiM/VAR by 0.25/0.28 FID and popular diffusion models LDM/DiT by 1.52/0.19 FID, respectively. When transferring our 1.0B model to the ImageNet 512×512 benchmark in a zero-shot manner, FlexVAR achieves competitive results compared to the VAR 2.3B model, which is a fully supervised model trained at 512×512 resolution.

1 Introduction

Autoregressive (AR) models aim to learn the probability distribution of the next token, offering great flexibility by generating tokens of any length. This design brings significant advancements in the field of Natural Language Processing (NLP), demonstrating satisfactory generality and transferability [4, 30, 31]. Concurrently, the computer vision field has been striving to develop large autoregressive models [28, 27, 1, 43, 29, 38]. These models employ visual tokenizers to discretize images into a series of 1D tokens [34, 22, 49, 52, 41] or 2D scales [44, 51, 42, 37, 25] and then utilize AR to model the next unit. However, these image autoregressive models typically output images at a single resolution, **the flexibility of AR has not yet been realized.**

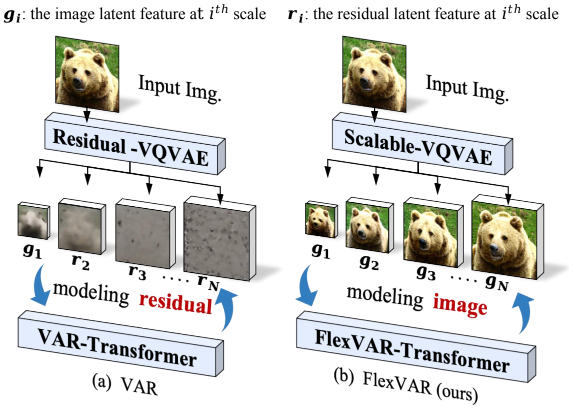

Figure 2: Comparison between VAR [44] and our FlexVAR. VAR predicts the GT[1] in step 1 and then predicts the residuals relative to the GT in all subsequent steps. Our FlexVAR predicts the GT at each step.

Recently, in image generation, VAR [44] has pioneered scale-wise autoregressive modeling, completing image autoregression based on 2D sequences. This approach predicts the next scale rather than the next token, thereby preserving the 2D structure of images and mitigating the issue of limited receptive fields in 1D causal transformers. Specifically, VAR predicts the ground-truth (GT)[1] of the smallest scale in the first step. Subsequently, at each step, it predicts the residuals of the current scale and the prior one. Finally, the outputs of each scale are upsampled to a uniform size and undergo weighted summation to generate the final output, as illustrated in Fig. 2(a). Successors [51, 42, 37, 25] have all adopted the residual design, assuming it to be effective. Although this technique achieves commendable performance, it encounters a primary challenge: The residual prediction relies on a rigid step design, restricting the flexibility to generate images with varying resolutions and aspect ratios, thus limiting the adaptability and flexibility of image generation. Meanwhile, residuals at different scales often lack semantic continuity, and this implicit prediction approach may limit the model's capacity to represent diverse image variations.

In this work, we examine the necessity of residual prediction in visual autoregressive modeling. Our intuition is that, in scale-wise autoregressive modeling, the ground-truth value of the current scale can be reliably estimated from the prior series of scales, rendering residual prediction (*i.e.*, predicting the bias between the current scale and the preceding one) unnecessary. Notably, predicting GT ensures semantic coherence between adjacent scales, making it more conducive for modeling the probability distribution of the scale. Additionally, this structure can output reasonable results at any step, breaking the rigid step design of the residual prediction and endowing autoregressive modeling with great flexibility.

Motivated by this, we systematically design the paradigm of visual autoregressive modeling without residual prediction, referred to as FlexVAR. Within FlexVAR, the ground-truth is predicted at each step instead of the residuals. Specifically, we design a scalable VQVAE tokenizer with multi-scale constraints, enhancing the VQVAE's robustness to various latent scales and thereby enabling image reconstruction at arbitrary resolutions. Then, the FlexVAR Transformer learns the probability distribution of a series of multi-scale latent features, modeling the ground-truth of the next scale, as shown in Fig. 2(b). Additionally, we propose scalable 2D positional embeddings, which incorporate

[1] To avoid confusion, we use ground-truth (GT) to represent image latent feature, and residual to represent residual latent feature.

2D learnable queries initialized with 2D sin-cosine weights. This approach enables the scale-wise autoregressive modeling to be extended to various resolutions/steps, including those beyond the resolutions/steps used during training, as shown in Fig. 1.

In a nutshell, this non-residual modeling approach ensures continuous semantic representation between adjacent scales. Simultaneously, it avoids the rigid step design inherent in residual prediction, significantly expanding the flexibility of image generation. FlexVAR can (1) generate images of various resolutions and aspect ratios, even exceeding the training resolutions; (2) support image-to-image tasks such as in/out-painting, image refinement, and image expansion without the need for fine-tuning; (3) enjoy flexible inference steps, allowing for accelerated inference with fewer steps or improved image quality with more steps.

2 Related Work

VQ-VAE [46] introduces a groundbreaking two-stage image generation paradigm: (1) encoding the image into a latent space and quantizing it to the nearest code in a fixed-size codebook; (2) modeling the discretized code using PixelCNN [45], which predicts the probability distribution of each code in raster scan order. This two-stage paradigm has laid the foundation for many subsequent works.

Raster-scan Manner Building on the aforementioned foundation, [12, 33] perform autoregressive learning in latent space with Transformer architecture. VQVAE-2 [35] and RQ-Transformer [22] use extra scales or stacked codes for next-image-token prediction. These works further advance the field and achieve impressive results. Recently, [41, 26] utilize a GPT-style next-token-prediction strategy to achieve high-quality image generation. [17] further improves this paradigm by introducing a mixture of autoregressive models, while [23] incorporates Mamba structure [13] to accelerate image generation. [48, 53, 14] combine diffusion processes into autoregressive modeling to address the information loss caused by quantization, which potentially degrades the quality of generated images.

Random-scan Manner Masked-prediction models learn to predict masked tokens in a BERT-style manner [9, 16, 2]. They introduce a bidirectional transformer that predicts masked tokens by attending to unmasked conditions, thus generating image tokens in a random-scan manner. This approach enables parallel token generation at each step, significantly improving inference efficiency. Specifically, [6, 5] apply masked-prediction models in class-to-image and text-to-image generation, respectively. MagViT series [50, 29] adapts this approach to videos by introducing a VQVAE for both images and videos. NOVA [8] first predicts temporal frames and then predicts spatial sets within each frame to achieve high-quality image/video generation.

Scaling-scan Manner VAR [44] establishes a new generation paradigm that redefines autoregressive learning on images from next-token-prediction to next-scale-prediction. VAR in parallel predicts image tokens at one scale, significantly reducing the number of inference steps. Following VAR, VAR-CLIP [51] achieves text-to-image generation by converting the class condition token into text tokens obtained from the CLIP. In terms of operational efficiency, [7] introduces an efficient decoding strategy, [37] incorporates linear attention mechanisms to accelerate image generation, and [25] designs a lightweight image quantizer, significantly reducing training costs. Regarding generation quality, [42, 36] optimize image details by using continuous tokenizers in combination with flow matching or diffusion model. Infinity [15] redefines the visual autoregressive model under a bitwise token prediction framework, remarkably enhancing generation capability and detail.

3 Methodology

3.1 Scale-wise Autoregression (Preliminary)

Scale-wise autoregressive models tokenize the input image into a sequence of multi-scale discrete image token maps $T = \{t_1, t_2, ..., t_n\}$, where t_i is the token map with the resolution of $h_i \times w_i$ downsampling from $t_n \in \mathcal{R}^{h_n \times w_n}$. Each autoregressive step generates an entire token map, rather than a single token. Compared to next-token-prediction, which contains one token at each step, t_i contains $h_i \times w_i$ tokens and is able to maintain the 2D structure.

Previous approaches [44, 51, 7, 37, 25, 42] typically follow a residual prediction paradigm. They only regress the ground-truth at the first scale (g_1), while at subsequent i^{th} scale, the residual between the preceding scale (g_{i-1}) and current scale (g_i) is predicted. We formulate residuals ($\{r_i\}_{i=2}^{n}$) as:

$$r_i = g_i - \mathrm{Upsample^i}(g_{i-1}), \tag{1}$$

here $\mathrm{Upsample^i}$ represents upsample g_{i-1} to the i^{th} scale. The autoregressive likelihood is:

$$p(g_1, r_2, \dots, r_n) = \prod_{i=1}^{n} p(r_i \mid g_1, r_2, \dots, r_{i-1}) \tag{2}$$

attention mechanisms (*e.g.*, Transformer [47]) are utilized to instantiate this modeling. During the i^{th} autoregressive step, all preceding residuals are merged in the autoregressive model, which then predicts the probability distribution of r_i. The $h_i \times w_i$ tokens in r_i are generated in parallel, conditioned on all preceding units. Thus, image token maps can be redefined as: $T = \{g_1, r_2, r_3..., r_n\}$. Finally, each image token map in T is upsampled to $\mathcal{R}^{h_n \times w_n}$ and summarized for image generation.

3.2 Overview of FlexVAR

Our FlexVAR is a flexible visual autoregressive image generation paradigm that allows autoregressive learning with ground-truth prediction rather than residual, enabling to generate reasonable images at any step independently. Within our approach: (1) A scalable VQVAE tokenizer quantizes the input images into tokens at various scales and reconstructs images, as detailed in Sec. 3.3. (2) A FlexVAR transformer is trained via scale-wise autoregressive modeling, with the removal of residuals, as detailed in Sec. 3.4.

3.3 Quantize & reconstruct images at various scales

Mainstream VQVAE tokenizers perform well at a single resolution. However, when scaling the latent space, they often fail to reconstruct images (as shown in Fig. 3). This observation motivates us to explore a scalable tokenizer that quantizes input images into tokens at various scales and reconstructs images. Specifically, the proposed scalable tokenizer first encodes an image into multi-scale latent space, and then uses a quantizer to convert latent space features into discrete tokens, finally a decoder is used to reconstruct the original images from the discrete tokens at each scale.

Encoding. Given an input image $I \in \mathcal{R}^{H \times W}$, an autoencoder $\mathcal{E}(\cdot)$ [12] is used to convert I into latent space f:

$$f = \mathcal{E}(I), \;\; f \in \mathcal{R}^{C \times h \times w} \tag{3}$$

here $h = \frac{H}{16}, w = \frac{W}{16}$. We then downsample f at K random scales to obtain multi-scale latent features $\mathcal{F} = \{f_1, f_2, ..., f_K\}$. f_k represents represents the latent feature of the k^{th} downsample from f. f_K matches the original resolution of f.

Quantizing. The quantizer $\mathcal{Q}(\cdot)$ includes a codebook $Z \in \mathcal{R}^{V \times C}$ containing V learnable vectors. The quantization process $q = \mathcal{Q}(f)$ is implemented by finding the Euclidean nearest code $q^{(k,i,j)}$ of each feature vector $f^{(k,i,j)}$ in multi-scale latent features $\mathcal{F}$:

$$q^{(k,i,j)} = \left(\mathrm{argmin}_{v \in [V]} \|\mathrm{Select}(Z, v) - f^{(k,i,j)}\|_2\right) \in [V] \tag{4}$$

where $\mathrm{Select}(Z, v)$ denotes selecting the v^{th} vector in codebook Z. Based on $\mathcal{F}$, we extract all $q^{(k,i,j)}$ and minimize the distance between q and f to train the quantizer $\mathcal{Q}$.

Decoding. The multi-scale images $\hat{\mathcal{I}} = \{\hat{I_1}, \hat{I_2}, ..., \hat{I_K}\}$ are reconstructed using the decoder $\mathcal{D}(\cdot)$ [12] given $q^{(k,i,j)}$. We follow Llamagen [41] to adopt the same loss functions ($\mathcal{L}_{vae}$) to train $\{\mathcal{E}, \mathcal{Q}, \mathcal{D}\}$ at each scale without special design. Therefore, the final loss function can be formulated:

$$\mathcal{L} = \sum_{k=1}^{K} \mathcal{L}_{vae}\left((I_k, \hat{I_k}), (f_k, q_k)\right) \tag{5}$$

3.4 Visual autoregressive modeling without residual

We reconceptualize the next-scale-prediction progress from residual prediction to GT prediction. As illustrate in Fig. 2 (b). Here, each autoregressive step predicts the GT of current scale, rather than the residual. We start by sampling N multi-scale token maps $\{g_1, g_2, \dots, g_N\}$ from latent feature f, each at an increasingly higher resolution $h_n \times w_n$, culminating in g_N matches the original feature map's resolution $\mathcal{R}^{C\times h\times w}$. The autoregressive likelihood is reformulated as:

$$p(g_1, g_2, \dots, g_n) = \prod_{i=1}^{n} p(g_i \mid g_1, g_2, \dots, g_{n-1}). \tag{6}$$

During the i^{th} autoregressive step, $g_i \in \mathcal{R}^{h_i \times w_i}$ contains $h_k \times w_k$ tokens are generated in parallel, conditioned on all preceding scales $\{g_1, g_2, \dots, g_{i-1}\}$.

Scalable Position Embedding. VAR utilizes fix-length Position Embedding (PE) by adding learnable queries to each step and h-w coordinates. This requires both training and inference to follow a fixed number of steps and resolutions, which limits the flexibility of the autoregressive process.

In our FlexVAR, we design a 2D scalable PE ($\mathcal{P} \in \mathcal{R}^{d\times 2h\times 2w}$) adding to the h-w coordinates. It contains $2h \times 2w$ learnable queries with d channels. At the ith step, $\mathcal{P}$ is upsampled/downsampled to match the scale of g_i. To ensure stability during linear interpolation across various scales, we set $\mathcal{P}$ to $2\times$ the size of the max latent space in traning. $\mathcal{P}$ is initialized using 2D sin-cosine PE [11] to ensure the 2D positional correlation. Additionally, we experimentally find that in our ground-truth prediction paradigm, incorporating PE for step embeddings is unnecessary (Tab. 6). Therefore, we remove the step embeddings to ensure the flexibility of steps in autoregressive modeling.

Step sampling. During training, we randomly sample the scale size in each step to enhance FlexVAR's capability to perceive any scale. Specifically, we set the maximum number of steps to 10, fixing the scale size of the first step to 1×1 and the last step to 16×16 (corresponding to 256×256 input images), and randomly sampling the scale sizes for the intermediate steps. Each step is dropped with a 5% probability, with a maximum of 4 steps being dropped. Thus, the number of steps during training is from 6 to 10. During inference, we use a default of 10 steps: {1, 2, 3, 4, 5, 6, 8, 10, 13, 16} (same as VAR). Our experimental results show more steps yield better performance (Fig. 6).

4 Experiments

4.1 Implementation details

FlexVAR tokenizer. Our scalable VQVAE tokenizer is configured with a downsampling factor of 16 and is initialized with the pre-trained weights from LlamaGen [41], the codebook size is set to 8912, and the latent space dimension is set to 32. The quantization of each scale shares the same codebook. We follow the VQVAE training recipe of LlamaGen. The training is on OpenImages [21] with a constant learning rate of 10^{-4}, AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay = 0.05, a batch size of 128, and for 20 epochs. K is set to 5 by default, indicating that each latent space is randomly sampled into 5 different resolutions.

FlexVAR transformer. We provide FlexVAR in three scales, with detailed configurations for each scale provided in Tab 1. FlexVAR is trained on the ImageNet-1K 256×256 using 80GB A100 GPUs. The training process employs the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and a weight decay rate of 0.05. The learning rate is set to 1e-4, with the training epochs varying between 180 and 350 depending on the model scale.

Model name	Layers	Params.	Heads	Dims.	Epoch
FlexVAR-*d*16	16	310M	16	1024	180
FlexVAR-*d*20	20	600M	20	1280	250
FlexVAR-*d*24	24	1.0B	24	1536	350

Table 1: Configuration of FlexVAR.

4.2 Overall Comparison

We compare FlexVAR with existing generative methods on the ImageNet-1K benchmark, including GAN, diffusion models, random-scan, raster-scan, and scaling-scan autoregressive models. As shown in Tab. 2. To ensure a fair comparison, we only present models with a size smaller than 1B. Our FlexVAR achieves state-of-the-art performance in all generative methods, and performs remarkably well compared to the VAR counterparts. Specifically, we achieve -0.45, -0.56, and -0.12 FID improvement compared with VAR at different model sizes.

Table 2: **Generative model comparison on class-conditional ImageNet 256×256**. Metrics include Fréchet inception distance (FID), inception score (IS), precision (Pre) and recall (rec). Step: the number of model runs needed to generate an image. Time: the relative inference time of VAR-$d30$ [44]. We present models with a size $\leq$ 1B.

Model	FID↓	IS↑	Pre↑	Rec↑	Param	Step	Time
Generative Adversarial Networks (GAN)							
BigGAN [3]	6.95	224.5	0.89	0.38	112M	1	–
GigaGAN [20]	3.45	225.5	0.84	0.61	569M	1	–
StyleGan-XL [40]	2.30	265.1	0.78	0.53	166M	1	0.2
Diffusion Models							
ADM [10]	10.94	101.0	0.69	0.63	554M	250	118
CDM [18]	4.88	158.7	–	–	–	8100	–
LDM-4-G [39]	3.60	247.7	–	–	400M	250	–
DiT-L/2 [32]	5.02	167.2	0.75	0.57	458M	250	2
DiT-XL/2 [32]	2.27	278.2	0.83	0.57	675M	250	2
Random-scan Manner (Mask Prediction)							
MaskGIT [6]	6.18	182.1	0.80	0.51	227M	8	0.4
RCG (cond.) [24]	3.49	215.5	–	–	502M	20	1.4
Raster-scan Manner (Token-wise Autoregressive)							
VQGAN-re [12]	18.65	80.4	0.78	0.26	227M	256	7
RQTran. [22]	13.11	119.3	–	–	821M	68	-
LlamaGen-XL [41]	2.62	244.08	0.80	0.57	775M	256	27
AiM [23]	2.56	257.2	0.81	0.57	763M	256	12
Scaling-scan Manner (Scale-wise Autoregressive)							
VAR-$d16$ [44]	3.55	280.4	0.84	0.51	310M	10	0.2
FlexVAR-$d16$	3.05	291.3	0.83	0.52	310M	10	0.2
VAR-$d20$ [44]	2.95	302.6	0.83	0.56	600M	10	0.3
FlexVAR-$d20$	2.41	299.3	0.85	0.58	600M	10	0.3
VAR-$d24$ [44]	2.33	312.9	0.82	0.59	1.0B	10	0.5
FlexVAR-$d24$	2.21	299.1	0.83	0.59	1.0B	10	0.5
FlexVAR-$d24$	2.08	315.7	0.83	0.59	1.0B	13	0.6

4.3 Zero-shot Comparison

Zero-shot inference with more steps. We use 13 steps for image generation without training, as shown in the last row of Tab. 2. FlexVAR can flexibly adopt more steps to improve image quality. By using 13 inference steps, FlexVAR further enhances the performance to 2.08 FID and 315 IS, manifesting strong flexibility and generalization capabilities. The specific steps design is detailed in the Supplementary Material.

Zero-shot inference on ImageNet 512×512 benchmark. We use FlexVAR-$d24$ to generate 512×512 images and evaluate on ImageNet-512 benchmark without training, as shown in Tab. 3. Surprisingly, our FlexVAR-$d24$ exhibits competitive performance when compared to VAR, despite FlexVAR being trained only on resolutions $\leq$ 256×256 and having only 1.0B parameters.

Model	Training Free	FID	IS	Params.
BigGAN [3]	×	8.43	177.9	112M
ADM [10]	×	23.24	101.0	554M
DiT-XL/2 [32]	×	3.04	240.8	675M
MaskGIT [6]	×	7.32	156.0	227M
VQGAN [12]	×	26.52	66.8	1.4B
VAR-$d36$ [12]	×	2.63	303.2	2.3B
FlexVAR-$d24$ (ours)	✓	4.43	314.4	1.0B

Table 3: Zero-shot inference on ImageNet 512×512 conditional generation. **Training Free** indicates whether the model is trained at the 512×512 resolution.

4.4 Ablation study

We conduct ablation studies on various design choices in FlexVAR . Due to the limited computational resources, we report the results trained with a short training scheme in Tab. 4, 5, 6, *i.e.*, 40 epochs (∼ 70K iterations).

Component-wise ablations. To understand the effect of each component, we start with standard VAR and progressively add each design, as shown in Tab. 4:

- **Baseline:** VAR uses a residual prediction paradigm, exhibits decent performance (1^{st} result), but its flexibility in image generation does not meet expectations (as described in Sec. 1).
- **Prediction type:** It is infeasible to directly convert the prediction type to GT, as seen in the 2^{nd} and 3^{rd} results. We employ the VQVAE tokenizers from VAR and Llamagen, both of which yield inferior performance. This is not surprising, as the current tokenizers lack robustness to images with varying latent space, while we force these tokenizers to obtain multi-scale latent features during training (we provide a detailed analysis in Fig. 3).
- **Tokenizer:** Our scalable tokenizer obtains reasonable multi-scale latent features during training, resulting in an improvement of -13.87 FID (the 4^{th} results). However, flexible image generation is not accomplished yet.
- **Position embedding:** As shown in the last result in Tab. 4, the introduction of our scalable Position Embedding (PE) provides high flexibility for image generation, and further enhances the performance to 3.71 FID.

Pred. type	VQVAE	PE	FID	IS
Residual	VAR	fixed-length	4.00	226.04
GT	VAR	fix-length	N/A	N/A
GT	Llamagen	fix-length	17.75	234.12
GT	ours	fix-length	3.82	229.35
GT	ours	scalable	3.71	230.22

Table 4: Ablation of diverse designs. We use the *next-scale-prediction* paradigm, explore the effects of different prediction types (residual/GT), VQVAE tokenizers (Llamagen/VAR/ours), and positional embedding (fix-length/scalable). N/A denotes the model does not converge during training. We report the results with model scale -$d20$ trained 40 epochs on ImageNet-1K.

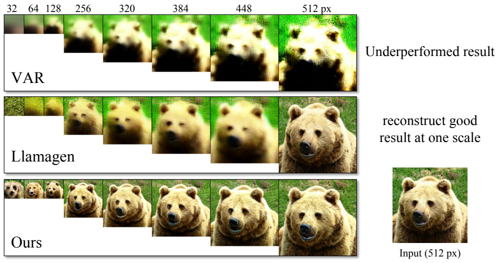

Figure 3: Compared with VQVAE [44, 41] for multi-scale reconstructing images, we downsample the latent features in VQVAE to multiple scales and use the VQVAE Decoder to reconstruct images. We upsample images < 100 pixels using bilinear interpolation for a better view.

Reconstruct images with different VQVAEs. In Fig. 3, we reconstruct multi-scale images by scaling the latent features in VQVAE tokenizers. Existing VQVAE tokenizers typically do not support scaling the latent features across a range of small to large scales. VAR's VQVAE [44] uses a residual-based training recipe, directly applying it to non-residual image reconstruction does not yield the anticipated results (the 1^{st} row). The VQVAE tokenizer from Llamagen [41] shows excellent reconstruction performance only at the original latent space, indicating that it is not feasible for scale-wise autoregressive modeling (the 2^{nd} row).

Depth	Atten. type	FID	IS	Params.	Time
-$d16$	Transformer	4.32	209.87	310M	0.2
	Mamba	4.22	200.04	370M	0.2
-$d20$	Transformer	3.71	230.22	600M	0.3
	Mamba	3.80	216.45	700M	0.3

Table 5: Ablation of the Mamba architectural. Models are trained 40 epochs ($\sim$ 70K iterations).

Step	h-w coordinates	learnable	FID	IS
fix-length	fixed-length	True	3.82	229.35
×	fixed-length	True	3.87	224.25
×	scaleable	False	3.74	224.04
×	scaleable	True	3.71	230.22

Table 6: Ablation of Position Embedding. × denotes that the corresponding PE is removed.

Transfer FlexVAR to Mamba. Recent work, AiM [23], uses the Mamba architecture for token-wise autoregressive modeling. Inspired by this, we modify FlexVAR with Mamba to evaluate the performance (Tab. 5). With similar model parameters, Mamba demonstrates competitive results compared to transformer models, indicating the GT prediction paradigm can effectively adapt to linear attention mechanisms like Mamba. However, considering that this Mamba architecture does not reflect the speed advantage, we do not integrate Mamba into our final version.

Mamba's inherent unidirectional attention mechanism prevents image tokens from achieving global attention within the same scale. To address this issue, we employ 8 scanning paths in different Mamba layers to capture global information. The specific Mamba architecture is detailed in the Supplementary Material.

Position Embedding. In Tab. 6, we experiment with several types of step PE and x-y coordinate PE. To make the model robust to inference steps and enable it to generate images at any resolution, we remove the fixed-length step embedding (results in the second row), and the performance showed only slight changes. We adopt a non-parametric variant, similar to ViT [11], which shows a 0.03 FID difference compared to the learnable variant.

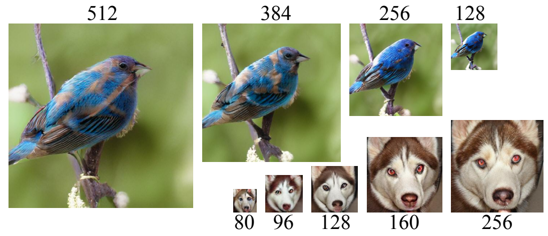

Figure 4: Generated samples from 80px to 512px. FlexVAR demonstrates strong consistency across various scales and can generate 512px images, despite the model being trained only on images $\leq$ 256px.

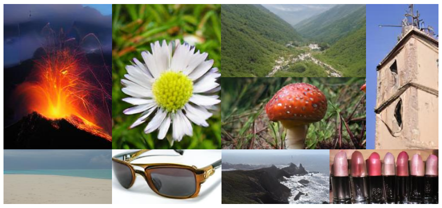

Figure 5: Generated samples with various aspect ratios. FlexVAR-$d24$ is used. FlexVAR demonstrates good visual quality across images with various aspect ratios.

VAE Performance Comparison. We compared the reconstruction performance of VAR and our proposed VAE in Tab. 7. The image reconstruction quality is measured by r-FID, reconstruction-FID on ImageNet validation set. We train FlexVAR-VAE with fewer epochs and smaller batch sizes. We observe its reconstruction quality inferior to VAR-VAE. Thus the improvement is not due to the quality of the discrete tokens, using a more robust FlexVAR-VAE might further improve the quality of generated images

	Epoch	**Batch-Size**	**rFID**
VAR-VAE	20	768	1.92
FlexVAR-VAE (ours)	10	128	3.79

Table 7: VAE Performance.

4.5 Analysis and Discussion

Generate images at various resolution. We show generated images at different resolutions using FlexVAR-$d24$ in Fig 1, 4. By controlling the inference steps, our FlexVAR can generate images at any resolution, despite being trained only on images with resolutions $\leq$ 256px. The generated images demonstrate strong semantic consistency across multiple scales, and the higher resolutions exhibit more detailed clarity. See the Supplementary Material for more zero-shot high-resolution generation samples and step designs.

Generate images at various ratio. We use FlexVAR-$d24$ to generate samples with various aspect ratios in Fig. 1 and Fig. 5. By controlling the aspect ratio at each step of the inference process, our FlexVAR allows for generating images with various aspect ratios, demonstrating the flexibility and controllability of our GT prediction paradigm. We control the height and width at each scale through approximate rounding. *e.g.*, to generate an image of size $H \times W$, the corresponding VAE latent feature size is $h \times w$, where $h = \frac{H}{16}$ and $w = \frac{W}{16}$. We adopt VAR's default set of 10 steps ($K = \{1, 2, 3, 4, 5, 6, 8, 10, 13, 16\}$) to determine the size at each scale. As a result, the $H \times W$ image corresponds to ten scales with sizes $\{\text{int}(h \times \frac{i}{16}),\ \text{int}(w \times \frac{i}{16})\}_{i \in K}$.

Generate images at various step. In Fig. 6, we investigate the FID and IS for generating 256×256 images from 6 to 16 steps with 3 different sizes (depth 16, 20, 24). As the number of steps increases, the quality of the generated images improves. The improvement is more significant in larger models (*e.g.*, FlexVAR-$d24$), as larger transformers are thought able to learn more complex and fine-grained image distributions. During training, we use up to 10 steps to avoid OOM (out-of-memory) problem. Surprisingly, in the inference stage, using 13 steps results in a performance gain of -0.13 FID. This observation indicates that our FlexVAR is flexible with respect to inference steps, allowing for fewer steps to speed up image generation or more steps to achieve higher-quality images. The details of various step designs are provided in the Supplementary Material.

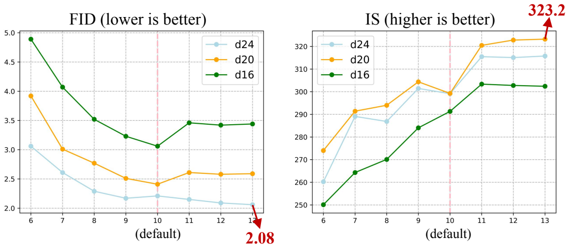

Figure 6: Zero-shot image generation at different steps (from 6 to 13 steps). FID and IS are used for evaluation. We use $\leq$ 10 steps for training, and FlexVAR can zero-shot transfer to 13 steps during inference and achieve better results.

Refine image at high resolution. In Fig. 7, we input low-resolution images (e.g., 256px×256px) and enable FlexVAR-$d24$ to output high-resolution refined images. Despite being trained only on $\leq$ 256px images, FlexVAR effectively refines image details by increasing the input image resolution, such as the eyes of the dogs in the example. This demonstrates the high flexibility of FlexVAR in image-to-image generation.

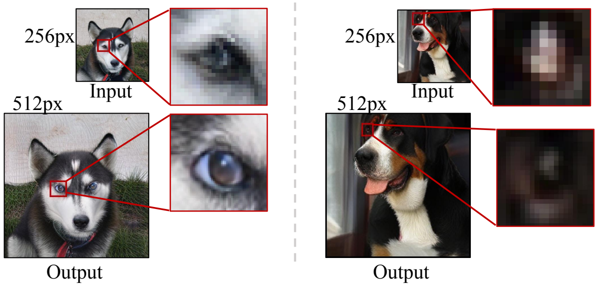

Figure 7: Zero-shot image refinement at high resolution. Zoom in for a better view.

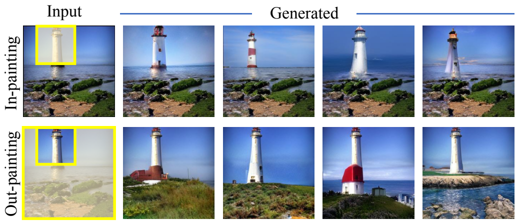

Figure 8: Zero-shot evaluation in/out-painting. The results show that FlexVAR can generalize to novel downstream tasks without special design and finetuning.

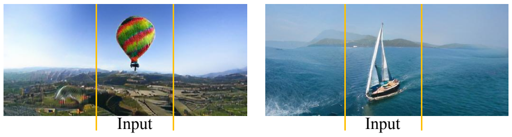

Figure 9: Zero-shot evaluation image expansion. The results show that FlexVAR can generalize to novel downstream tasks without special design and fine-tuning.

Image in-painting and out-painting. For in-painting and out-painting, we teacher-force ground-truth tokens outside the mask and let the model only generate tokens within the mask. Class label information is also injected. The results are visualized in Fig. 8. Without modifications to the architecture design or training, FlexVAR achieves decent results on these image-to-image tasks.

Image extension. For image extension, we extend images with an aspect ratio of 1:2 for the target class, with the ground-truth tokens forced to be in the center. The results are visualized in Fig. 9. FlexVAR shows decent results in image extension, indicating the strong generalization ability and flexibility of our architecture.

Failure case. FlexVAR fails to generate images with a resolution 3× or more than the training resolution, as illustrated in Fig. 10. These cases typically feature noticeable wavy textures and blurry areas in the details. This failure is likely due to the overly homogeneous structure of the current training dataset. *i.e.*, ImageNet-1K generally lacks multi-scale objects ranging from coarse to fine, leading to errors in generating details of high-resolution objects.

We hypothesize that training the model with a more complex dataset that includes images with fine-grained details, the model might become robust for higher resolutions.

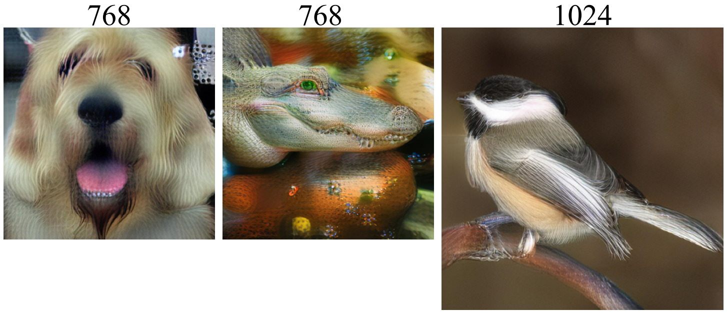

Figure 10: Failure cases at high resolutions (768 & 1024). FlexVAR shows wavy textures when generating high-resolution images. Zoom in for a better view.

5 Conclusion

In this paper, we introduce FlexVAR, a flexible visual autoregressive image generation paradigm that allows autoregressive learning without residual prediction. We design a scalable VQVAE tokenizer and FlexVAR-Transformer for this purpose. This ground-truth prediction paradigm endows the autoregressive model with great flexibility and controllability, enabling image generation at various resolution, aspect ratio, and inference step, beyond those used during training. Moreover, it can zero-shot transfer to various image-to-image generation tasks. We hope FlexVAR will serve as a solid baseline and help ease future research of visual autoregressive modeling and related areas.

Limitations. We observe that when generating images with a resolution $\geq 3\times$ larger than the training image, noticeable wavy textures appear (Fig. 10). This issue may be attributed to the homogeneous structure of the ImageNet-1K training set. We will investigate this further in future work to explore how to ensure stability in zero-shot image generation at higher resolutions.

Broader Impact. This research strictly follows established practices for class-to-image (c2i) model training and evaluation. Similar to most generative models, our approach may inherit biases present in the training datasets. We advocate for the responsible use of this technology and caution when deploying it in real-world scenarios.

Acknowledgments and Disclosure of Funding

This work is supported by the National Natural Science Foundation of China (No. 92470203, U23A20314, U24B20179, 6120106009), the Beijing Natural Science Foundation (No. L242022), and the Fundamental Research Funds for the Central Universities (No. 2024XKRC082).

References

[1] Yutong Bai, Xinyang Geng, Karttikeya Mangalam, Amir Bar, Alan L Yuille, Trevor Darrell, Jitendra Malik, and Alexei A Efros. Sequential modeling enables scalable learning for large vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22861–22872, 2024.

[2] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. arxiv 2018. *arXiv preprint arXiv:1809.11096*, 1809.

[4] 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. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] 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.

[6] 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*, pages 11315–11325, 2022.

[7] Zigeng Chen, Xinyin Ma, Gongfan Fang, and Xinchao Wang. Collaborative decoding makes visual auto-regressive modeling efficient. *arXiv preprint arXiv:2411.17787*, 2024.

[8] Haoge Deng, Ting Pan, Haiwen Diao, Zhengxiong Luo, Yufeng Cui, Huchuan Lu, Shiguang Shan, Yonggang Qi, and Xinlong Wang. Autoregressive video generation without vector quantization. *arXiv preprint arXiv:2412.14169*, 2024.

[9] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[11] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[12] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[13] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[14] Jiatao Gu, Yuyang Wang, Yizhe Zhang, Qihang Zhang, Dinghuai Zhang, Navdeep Jaitly, Josh Susskind, and Shuangfei Zhai. Dart: Denoising autoregressive transformer for scalable text-to-image generation. *arXiv preprint arXiv:2410.08159*, 2024.

[15] Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. *arXiv preprint arXiv:2412.04431*, 2024.

[16] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[17] Wanggui He, Siming Fu, Mushui Liu, Xierui Wang, Wenyi Xiao, Fangxun Shu, Yi Wang, Lei Zhang, Zhelun Yu, Haoyuan Li, et al. Mars: Mixture of auto-regressive models for fine-grained text-to-image synthesis. *arXiv preprint arXiv:2407.07614*, 2024.

[18] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

[19] Vincent Tao Hu, Stefan Andreas Baumann, Ming Gui, Olga Grebenkova, Pingchuan Ma, Johannes Fischer, and Björn Ommer. Zigma: A dit-style zigzag mamba diffusion model. In *European Conference on Computer Vision*, pages 148–166. Springer, 2024.

[20] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10124–10134, 2023.

[21] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International journal of computer vision*, 128(7):1956–1981, 2020.

[22] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11523–11532, 2022.

[23] Haopeng Li, Jinyue Yang, Kexin Wang, Xuerui Qiu, Yuhong Chou, Xin Li, and Guoqi Li. Scalable autoregressive image generation with mamba. *arXiv preprint arXiv:2408.12245*, 2024.

[24] Tianhong Li, Dina Katabi, and Kaiming He. Return of unconditional generation: A self-supervised representation generation method. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[25] Xiang Li, Kai Qiu, Hao Chen, Jason Kuen, Jiuxiang Gu, Bhiksha Raj, and Zhe Lin. Imagefolder: Autoregressive image generation with folded tokens. *arXiv preprint arXiv:2410.01756*, 2024.

[26] Dongyang Liu, Shitian Zhao, Le Zhuo, Weifeng Lin, Yu Qiao, Hongsheng Li, and Peng Gao. Lumina-mgpt: Illuminate flexible photorealistic text-to-image generation with multimodal generative pretraining. *arXiv preprint arXiv:2408.02657*, 2024.

[27] Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. Unified-io 2: Scaling autoregressive multimodal models with vision language audio and action. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26439–26455, 2024.

[28] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. In *The Eleventh International Conference on Learning Representations*, 2022.

[29] Zhuoyan Luo, Fengyuan Shi, Yixiao Ge, Yujiu Yang, Limin Wang, and Ying Shan. Open-magvit2: An open-source project toward democratizing auto-regressive visual generation. *arXiv preprint arXiv:2409.04410*, 2024.

[30] OpenAI. Introducing chatgpt. `https://openai.com/blog/chatgpt/`, 2022.

[31] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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

[33] 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.

[34] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

[35] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

[36] Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Flowar: Scale-wise autoregressive image generation meets flow matching. *arXiv preprint arXiv:2412.15205*, 2024.

[37] Sucheng Ren, Yaodong Yu, Nataniel Ruiz, Feng Wang, Alan Yuille, and Cihang Xie. M-var: Decoupled scale-wise autoregressive modeling for high-quality image generation. *arXiv preprint arXiv:2411.10433*, 2024.

[38] Zhongwei Ren, Yunchao Wei, Xun Guo, Yao Zhao, Bingyi Kang, Jiashi Feng, and Xiaojie Jin. Videoworld: Exploring knowledge learning from unlabeled videos. *arXiv preprint arXiv:2501.09781*, 2025.

[39] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[40] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. volume abs/2201.00273, 2022.

[41] 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.

[42] Haotian Tang, Yecheng Wu, Shang Yang, Enze Xie, Junsong Chen, Junyu Chen, Zhuoyang Zhang, Han Cai, Yao Lu, and Song Han. Hart: Efficient visual generation with hybrid autoregressive transformer. *arXiv preprint arXiv:2410.10812*, 2024.

[43] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

[44] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv preprint arXiv:2404.02905*, 2024.

[45] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016.

[46] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[47] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[48] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

[49] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021.

[50] Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. Magvit: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10459–10469, 2023.

[51] Qian Zhang, Xiangzi Dai, Ninghua Yang, Xiang An, Ziyong Feng, and Xingyu Ren. Var-clip: Text-to-image generator with visual auto-regressive modeling. *arXiv preprint arXiv:2408.01181*, 2024.

[52] Chuanxia Zheng, Tung-Long Vuong, Jianfei Cai, and Dinh Phung. Movq: Modulating quantized vectors for high-fidelity image generation. *Advances in Neural Information Processing Systems*, 35:23412–23425, 2022.

[53] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*, 2024.

NeurIPS Paper Checklist

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

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

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

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

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

IMPORTANT, please:

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

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction outline our contributions and scope.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Limitations are discussed in the **Conclusion** section.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [NA]

   Justification: This paper does not include theoretical assumptions and claims.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The detailed information about model designs and experimental settings in the paper makes it possible for researchers to reproduce the model with the same public dataset.

   Guidelines:

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

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The code will be released. The datasets are all public.

   Guidelines:

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

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The experimental setting is presented in the **Experiments** section.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: It is uncommon for top conference papers in this area to report error bars or similar statistical measures. Our paper aligns with this standard of other SOTA papers, which are the key criteria for evaluation in this domain.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: This paper provides sufficient information on the computer resources (type of GPU, inference time)

   Guidelines:

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

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conducted in the paper conforms to the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: This paper provides a discussion of potential societal impacts in the **Conclusion** section.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [No]

Justification: The datasets used are public datasets from existing papers.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: his paper cites the original papers for their code or datasets.

Guidelines:

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

13. **New assets**

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

Answer: [NA]

Justification: The paper does not release new assets

Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

Answer: [NA]

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

Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [NA]

    Justification: The paper does not involve research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology in this work.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Inference steps

In Tab. 8, we list the scales corresponding to different inference steps. The scales in each step are not fixed and can be flexibly adjusted during inference. Note that during training, we only limit the maximum number of steps to 10 and randomly sample the scale for each step, so the scales during the training process do not follow Tab. 8

Reso	Step	Scale
256px	6	{1, 2, 4, 6, 10, 16}
	7	{1, 2, 3, 5, 8, 11, 16}
	8	{1, 2, 3, 4, 6, 10, 13, 16}
	9	{1, 2, 3, 4, 5, 7, 10, 13, 16}
	10	{1, 2, 3, 4, 5, 6, 8, 10, 13, 16}
	11	{1, 2, 3, 4, 5, 6, 7, 9, 11, 13, 16}
	12	{1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 16}
	13	{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 16}
384px	11	{1, 2, 3, 4, 5, 6, 8, 10, 13, 16, 24}
512px	12	{1, 2, 3, 4, 5, 6, 8, 10, 13, 16, 23, 32}
	15	{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 16, 23, 32}

Table 8: Scale configurations of various inference steps.

B Extent visual autoregressive modeling with Mamba

Unlike attention mechanisms that utilize explicit query-key-value (QKV) interactions to integrate context, Mamba faces challenges in handling bi-directional interaction. Therefore, prior Mamba-based visual autoregressive work [37] only used Mamba to model the unidirectional relationship between scales, relying on additional Transformer layers to process tokens within one scale.

In this work, we adopt a composition-recomposition strategy to obtain global information in Mamba network. Specifically, we utilize a Zigzag scanning strategy [19] over the spatial dimension. We alternate between eight distinct scanning paths across different Mamba layers (as shown in Fig. 11), which include:

- (a) top-left to the bottom-right.
- (b) top-left to the bottom-right.
- (c) bottom-left to the top-right.
- (d) bottom-left to the top-right.
- (e) bottom-right to the top-left.
- (f) bottom-right to the top-left.
- (g) top-right to the bottom-left.
- (h) top-right to the bottom-left.

C Qualitative results with different steps.

In Fig. 12, we show some generated samples with {6, 8, 10, 12} steps. Our FlexVAR uses up to 10 steps for autoregressive modeling during training to avoid OOM (out-of-memory), while it can naturally transfer to any number of steps during inference. The samples generated with different steps are highly similar, differing only in some details. Generally, more steps result in better image details.

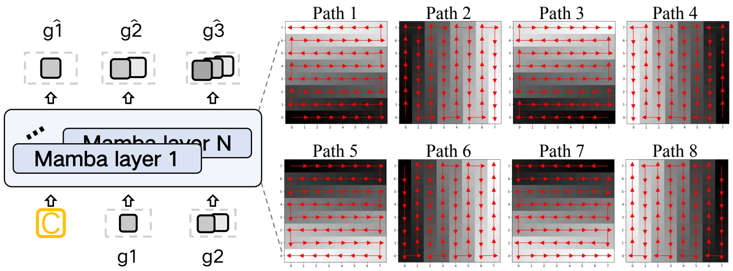

Figure 11: Sptial scan paths for Mamba.

D Qualitative results with various resolutions.

Fig. 13 shows some generated samples with {256, 384, 512} resolutions. FlexVAR uses up to 256×256 resolution images for training, it can generate images with higher resolutions such as 384 and 512. The generated images demonstrate strong semantic consistency across multiple scales, and the higher resolutions display more detailed clarity.

E Qualitative results with different VQVAE tokenizers.

Image reconstruction. We compare more image reconstruction results in Fig. 14. First, we encode the image into the latent space and performe multi-scale downsampling, then reconstruct the original image through the VQVAE decoder. It is evident that only our scalable VQVAE can perform image reconstruction at various scales.

Generate images with GT prediction. We visualize the generated samples with VQVAE tokenizers from VAR, Llamagen, and ours, corresponding to the 2^{nd}, 3^{rd} and 5^{th} results in Tab. 6 in the main paper. As shown in Fig. 15, the VAR tokenizer, trained with a residual paradigm, fails to generate images under GT prediction; the generation samples of Llamagen's tokenizer are not up to the mark, due to its discrete tokens at intermediate steps being suboptimal.

F Additional Visual Results.

We show more generated samples in Fig. 16.

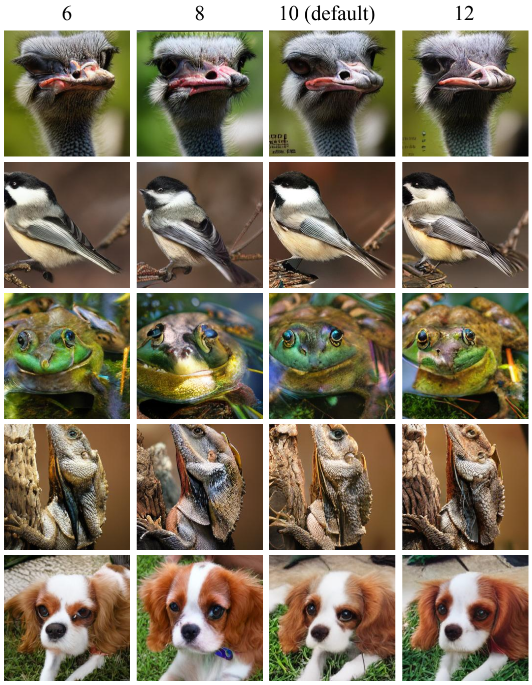

Figure 12: Some generated samples with {6, 8, 10, 12} steps. Note the model is trained with steps ≤ 10. More steps typically result in better image details. Zoom in for a better view.

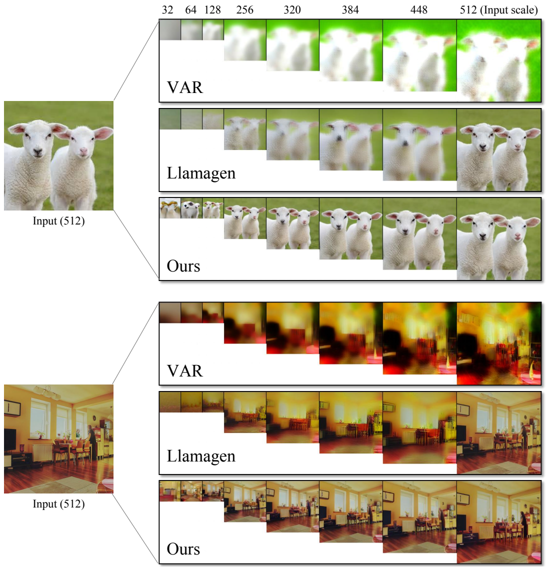

Figure 14: Compared with different VQVAE tokenizers [44, 41] for multi-scale reconstructing images, we downsample the latent features in VQVAE to multiple scales and then use the VQVAE Decoder to reconstruct images. We upsample images < 100 pixels using bilinear interpolation for a better view.

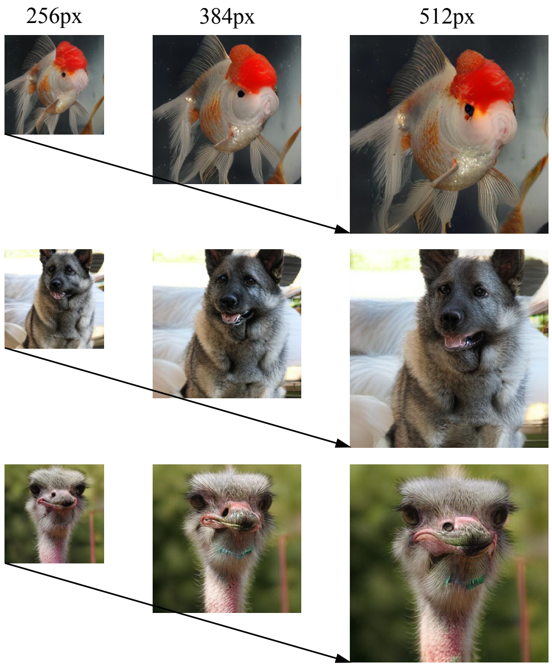

Figure 13: Some generated samples with {256, 384, 512} resolutions. Note the model is trained with a resolution of $\leq 256\times256$. Zoom in for a better view.

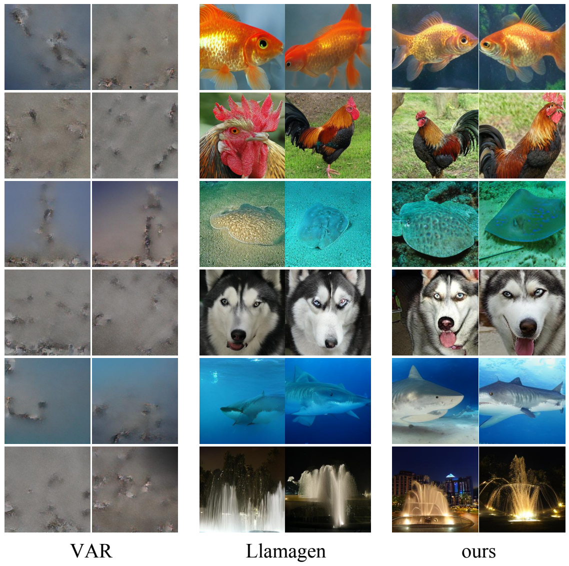

Figure 15: Some generated samples with different VQVAE tokenizers (Llamagen & VAR), corresponding to the 2^{nd} and 3^{rd} results in Tab. 6 in the main paper. We report the results with model scale -$d20$ trained 40 epochs ($\sim$ 70K iterations) on ImageNet-1K. Zoom in for a better view.

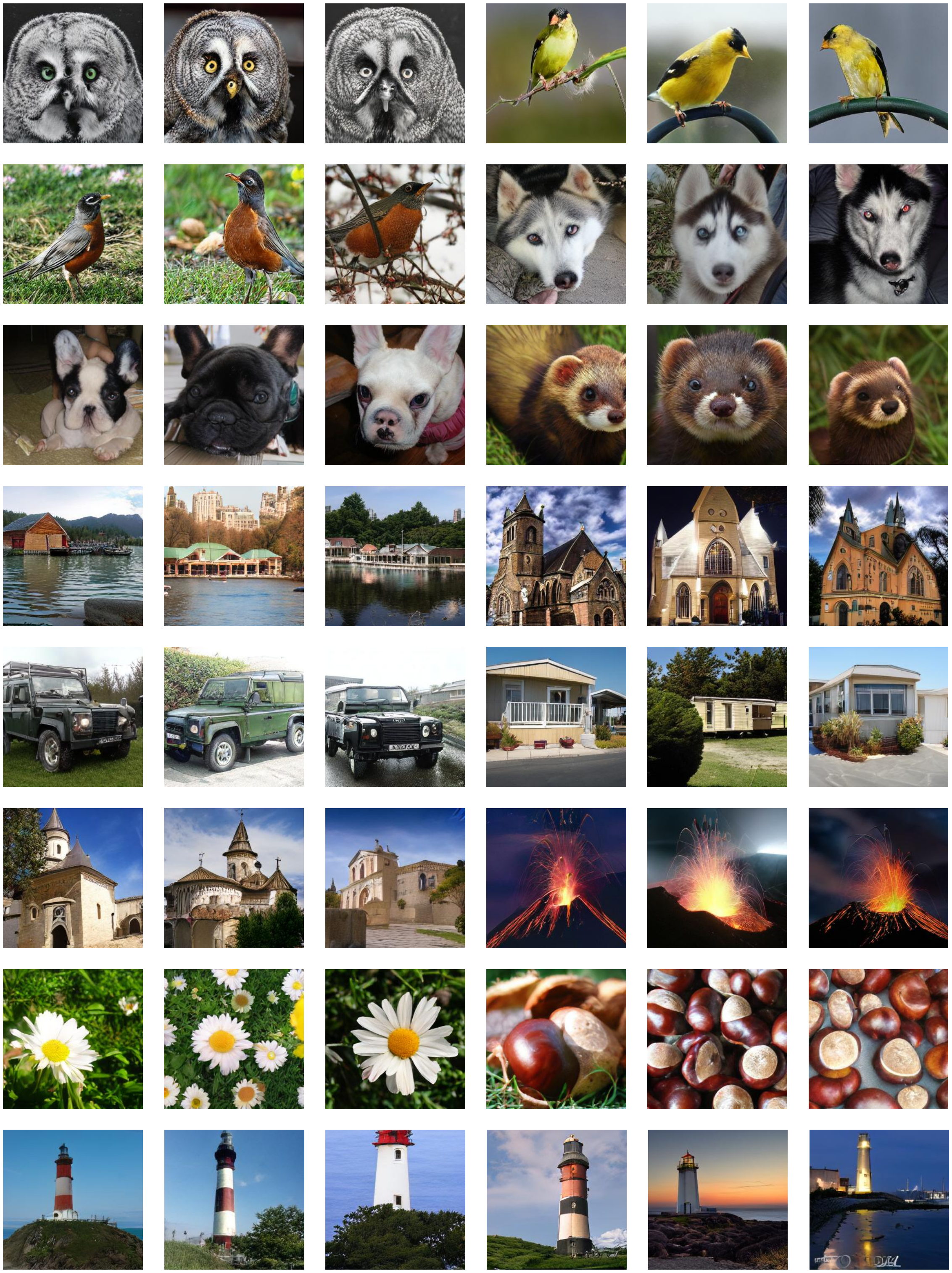

Figure 16: Some generated 256×256 samples.