Learning to Expand Images for Efficient Visual Autoregressive Modeling

Ruiqing Yang[1], Kaixin Zhang[2], Zheng Zhang[3], Shan You[4], Tao Huang[5*]

[1]University of Electronic Science and Technology of China
[2]School of Computer Science and Engineering, Central South University
[3]Xidian University   [4]SenseTime Research   [5]Shanghai Jiao Tong University

yrq@std.uestc.edu.cn, kaixinzhang@csu.edu.cn,
zheng.zhang@stu.xidian.edu.cn, youshan@sensetime.com, t.huang@sjtu.edu.cn

Abstract

Autoregressive models have recently shown great promise in visual generation by leveraging discrete token sequences akin to language modeling. However, existing approaches often suffer from inefficiency, either due to token-by-token decoding or the complexity of multi-scale representations. In this work, we introduce Expanding Autoregressive Representation (EAR), a novel generation paradigm that emulates the human visual system's center-outward perception pattern. EAR unfolds image tokens in a spiral order from the center and progressively expands outward, preserving spatial continuity and enabling efficient parallel decoding. To further enhance flexibility and speed, we propose a length-adaptive decoding strategy that dynamically adjusts the number of tokens predicted at each step. This biologically inspired design not only reduces computational cost but also improves generation quality by aligning the generation order with perceptual relevance.Extensive experiments on ImageNet demonstrate that EAR achieves state-of-the-art trade-offs between fidelity and efficiency on single-scale autoregressive models, setting a new direction for scalable and cognitively aligned autoregressive image generation.Code is available at https://github.com/RuiqingYoung/EAR.

1. Introduction

Inspired by the remarkable success of large language models (LLMs) [37] in natural language processing, their autoregressive next-token prediction strategy has been extended to the field of image generation, driving substantial progress in autoregressive visual modeling.

Recent autoregressive image generation methods follow the design of large language models (LLMs), where an image is first encoded into a sequence of discrete visual tokens
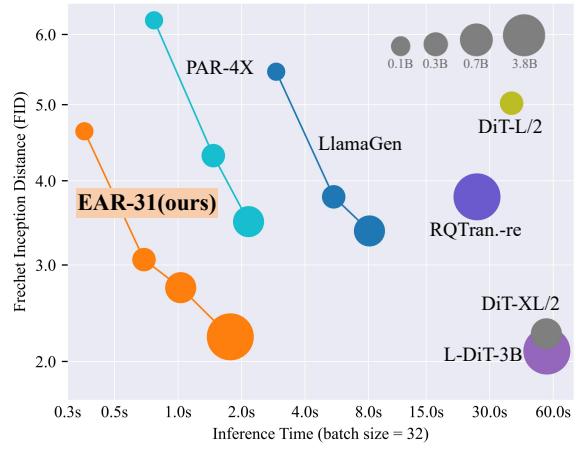

Figure 1. Scaling behavior of different generation methods on ImageNet 256×256 generation benchmark.

using a tokenizer such as VQ-VAE [35]. These tokens are then autoregressively generated one by one. However, since generating a single image typically requires a large number of tokens (e.g., 16×16), the next-token prediction paradigm leads to significantly more inference steps compared to diffusion models [10], resulting in much slower generation speed. Although several recent works such as PAR [39] and MAR [15] have attempted to accelerate autoregressive generation by patch-wise or randomized generation strategies, they often suffer from increased model complexity or limited flexibility. More recently, VAR achieves a significant reduction in inference steps through a next-scale prediction mechanism, attaining competitive performance in both accuracy and speed compared to diffusion models. However, it introduces substantial computational overhead during inference and requires training multi-scale VQ-VAE tokenizers. This raises an important question: *is there an image modeling paradigm that not only obeys the nature of image understanding to obtain promising generative capabilities,*
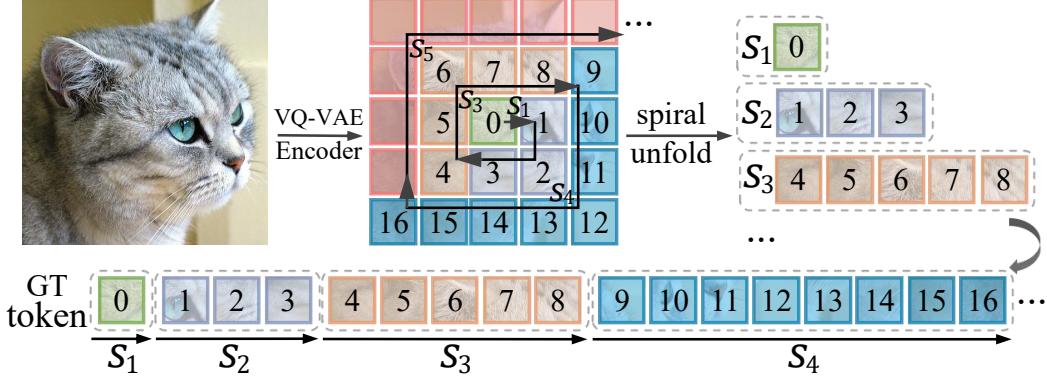

Figure 2. The proposed spiral unfolding and parallel generation strategy. The generation starts from the center of an image, and expands outwards spirally. Each S stands for a generation step.

but also retains the simplicity and efficiency of autoregressive generation?

In this paper, inspired by insights from cognitive neuroscience, we propose a novel image generation framework called Expanding Autoregressive Modeling (EAR). To emulate the human visual system's center-outward scanning mechanism, we introduce a novel spiral unfolding strategy that preserves the spatial continuity among image tokens. As illustrated in Figure 2, unlike the conventional raster-scan-based token arrangements, in our method, each token remains adjacent to its spatial neighbors after unfolding, enabling a center-to-outward generation process that more naturally aligns with the inherent spatial structure of images.

Meanwhile, to better allocate the generation resources, we design a progressive length-varying decoding strategy: in the challenging early stages, the model predicts fewer tokens to ensure a reliable start of the central region; as generation proceeds outward, the number of tokens per step increases accordingly to boost the efficiency. As a result, our EAR obtains several major advantages over previous methods:

- **Preserved spatial continuity:** Unlike multi-scale prediction methods (e.g., VAR) or patch-chunk generation strategies (e.g., PAR), spiral ordering avoids disrupting adjacency among visual tokens, thereby improving image fidelity and reducing local artifacts.
- **Efficient parallel expansion:** Initiating generation from the center and expanding outward enables controlled parallel prediction of multiple tokens per step, resulting in significantly faster inference while preserving generation quality.
- **Biologically inspired perceptual alignment:** The center-to-outward generation pattern aligns with human visual processing mechanisms, including central fixation bias and center–surround attention profiles [18] [2]), promoting cognitive plausibility in the generated images.

We implement a series of EAR model variants on 256×256 ImageNet generation. The results demonstrate that, our EAR achieves the optimal balance between image quality and generation speed, as summarized in Figure 1. For example, our EAR31-XL model achieved an FID of 2.7 and runs 8 times faster than LlamaGen with the same number of parameters.

2. Related Work

2.1. Autoregressive Models in Image Generation

Autoregressive models have been widely adopted in image generation by drawing parallels between language token modeling and visual data synthesis. Early studies [14, 26, 28, 41] propose the next-pixel prediction paradigm, where each pixel is treated as a token. These methods often rely on flattening 2D images into 1D token sequences via raster scan order [4, 5, 14, 34], and employ CNNs or Transformers to predict next pixel based on all previous pixels. In order to improve the quality and efficiency of image generation, VQVAE and VQGAN [9, 35] introduce the tokenizer that compresses image patch into discrete visual tokens. Building on this foundation, recent works [23, 38, 42] make substantial progress, achieving image generation quality on par with diffusion-based models [24, 25, 27]. For example, LLamaGen [31] redesigns the image tokenizer to support multiple downsampling ratios, resulting in higher image generation quality.

2.2. Acceleration of Autoregressive Models

In natural language processing [3, 6] and computer vision [8, 20, 33], the transformer [36] has become one of the most popular backbone networks, owing to its scalability and strong performance. However, this performance often comes with long inference latency, which poses a challenge for the efficient deployment. To alleviate this issue, many studies [21, 22, 29, 40, 43]propose acceleration techniques
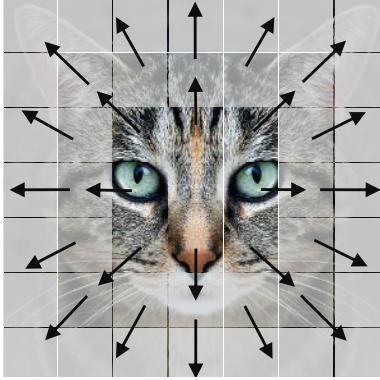
Figure 3. Illustration of human visual perception. The observations fixate initially at the center of an image, then gradually expand outwards.

to improve inference speed of transformer. Motivated by these efforts, some researchers [11, 16, 30] begin exploring acceleration strategies specifically for transformer-based autoregressive models. For instance, Anagnostidis et al.[1] introduce a context-aware pruning method that dynamically removes uninformative tokens during generation. DD [19] proposed a method based on knowledge distillation, which significantly reduced the inference steps of the autoregressive model. Despite acceleration efforts, most methods sacrifice quality, depend on task-specific heuristics, or lack generality across token structures and decoding schemes. Many also target language models and struggle with high-res image generation, where spatial coherence is vital. To address this, we propose EAR, a spatially aware framework with a biologically inspired center-outward generation and flexible next-any-tokens prediction. This enables parallel multi-token generation, speeding inference while preserving image quality and structure for efficient, high-fidelity synthesis.

3. Learning to Expand Images

3.1. Preliminary on Visual Autoregressive Modeling

Current visual autoregressive (AR) generation methods can be categorized based on the prediction type into next-token prediction and next-scale prediction.

3.1.1. Next-token Prediction

Autoregressive models decompose the joint distribution of a token sequence $x = (x_1, x_2, \ldots, x_T)$ into a product of conditional probabilities:

$$p(x_1, x_2, \ldots, x_T) = \prod_{t=1}^{T} p(x_t \mid x_1, x_2, \ldots, x_{t-1}) \quad (1)$$

This formulation assumes each token depends only on its

previous tokens. It has been widely successful in language modeling and has recently been applied to visual domains by tokenizing images into sequences (e.g., using raster-scan order). However, sequential token generation limits speed, particularly for high-resolution images.

3.1.2. Next-scale Prediction

To improve generation efficiency, the next-scale prediction paradigm introduces a hierarchical token generation process. Rather than modeling individual tokens, it organizes token maps across multiple spatial resolutions (scales) and generates them in a coarse-to-fine manner:

$$p(r_1, r_2, \ldots, r_S) = \prod_{s=1}^{S} p(r_s \mid r_1, \ldots, r_{s-1}) \quad (2)$$

Here, r_s denotes the token map at scale s, which can represent a downsampled version of the image. This approach allows the model to capture global structure first and then refine local details. In hierarchical models like VAR [16], generating a high-resolution image (e.g., 256×256) typically involves decoding multiple scales (e.g., 10 levels). Multi-scale token maps are concatenated and jointly input to the model, which increases computation due to longer token sequences and multi-scale attention. Additionally, all intermediate token maps need to be stored during generation, further increasing GPU memory consumption.

3.2. Human Cognition Inspired Visual Modeling

In this paper, we try to figure out two major questions: (1) What is the more natural and precise image understanding manner? (2) What parallelization of token generation is better in both quality and speed?

To answer the questions, we refer to human visual perception research in cognitive neuroscience (see an illustration in Figure 3 for easier understanding), where studies consistently show that observers tend to fixate initially at the center of an image—a phenomenon known as the central fixation bias [18]. Meanwhile, magnetoencephalographic (MEG) experiments reveal that attention in early visual cortex exhibits a center-surround profile, where focal enhancement at the center is surrounded by a narrow inhibitory zone, reflecting recurrent processing and feature binding during early feedforward stages [2]. This indicates that human perception expands progressively from the center to outside, with each expansion step depending on its surrounding regions.

These motivates us to build a more efficient image modeling manner by starting from the center, and gradually expanding outward. Meanwhile, a straightforward parallelization can be built inside this expansion process.

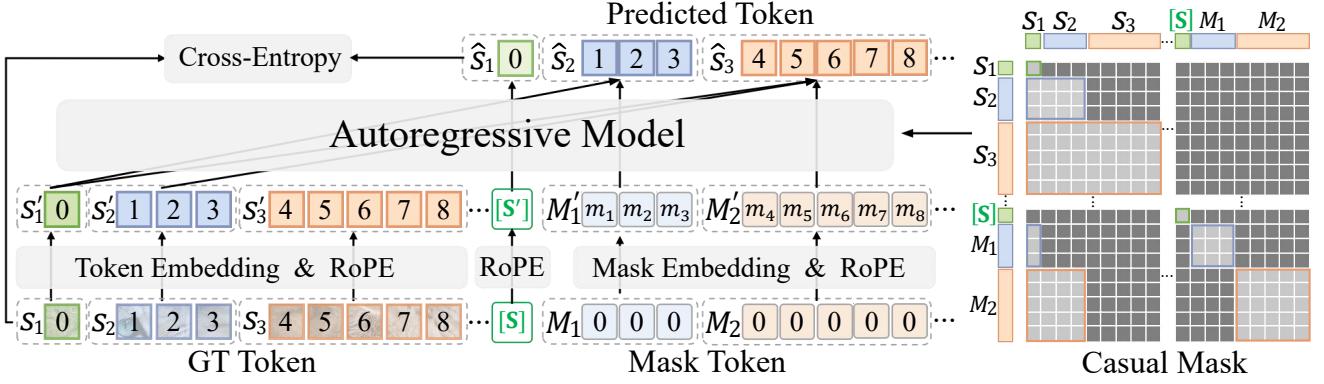Next, we will officially introduce the two cores of our

Figure 4. Training EAR transformer on tokens. Note that [S] denotes the start token derived from the class embedding. In practice, the Mask Token also attends to [S] through the attention mechanism to capture class information, although this detail is omitted in the figure for clarity.

method: **spiral token unfolding** and **next-any-tokens parallelization**.

3.2.1. Spiral Token Unfolding

Inspired by how the human visual system prioritizes fine-grained perception in the fovea and gradually expands attention to the periphery, we propose a novel token unfolding strategy called **spiral unfolding**, which preserves spatial continuity and enables center-outward image generation. Specifically, we start from the central token at $\left(\frac{n}{2}, \frac{n}{2}\right)$ and traverse the 2D token grid in a clockwise spiral (right $\rightarrow$ down $\rightarrow$ left $\rightarrow$ up), producing a *spiral index map* that defines the token order, as shown in Figure 2.
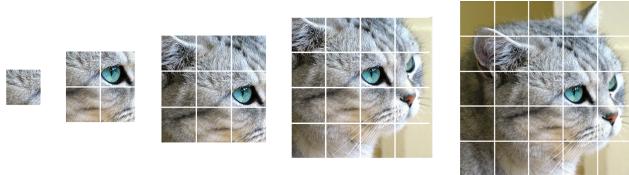
3.3. Next-Any-Tokens Parallelization

Figure 5. EAR generation process.

After applying spiral unfolding to the image tokens, we obtain a token sequence that expands outward from the image center. To enable the model to efficiently generate a complete image following the expansion pattern shown in Figure 5, we design the Expandable Autoregressive Modeling (EAR) framework, and introduce a novel prediction paradigm: **next-any-tokens prediction**.

During the training, as illustrated in Figure 4, we introduce a learnable embedding parameter similar to the mask token in MaskGit. This mask token is duplicated to match the spatial size of the ground truth (GT) tokens and then concatenated with the start token (obtained from the class embedding) and the GT tokens, as illustrated in Figure 4. Both GT tokens and mask tokens are encoded with the same rotary positional encoding (RoPE). The resulting sequence is then fed into the autoregressive model along with our designed EAR causal mask, which ensures that the mask tokens are autoregressively transformed into predicted tokens. The predicted tokens represent a probability distribution over the codebook entries in VQ-VAE for each token position. Finally, we compute the loss between the predicted tokens and the GT tokens to train our EAR model. Therefore, the next-any-tokens prediction in our EAR framework can be formulated as:

$$p(s_1, s_2, \ldots, s_T) = \prod_{t=1}^{T} p(m_t \mid s_1, s_2, \ldots, s_{t-1}) \quad (3)$$

where m_t denotes the mask tokens at the t-th step, and $s_1, s_2, \ldots, s_{t-1}$ are the ground truth tokens revealed in the previous steps.

3.3.1. Mask Design

To accommodate our next-any-tokens prediction paradigm, we design a customized causal mask. Specifically, for the mask tokens within the same generation step, we allow them to attend to all the ground truth (GT) tokens from previous steps, as well as to each other within the same step. This design prevents information leakage from future tokens while ensuring contextual consistency within the current step.

However, if no attention mask is applied to the GT tokens, the multi-layer structure of the Transformer may cause information from future GT tokens to be indirectly propagated to earlier-step GT tokens through residual connections and self-attention in deeper layers. To address this, we also apply attention masks to the GT tokens, restricting each GT token to only attend to GT tokens from the current and previous steps. An illustration of our EAR-specific causal mask design is shown in Figure 4 (right part).
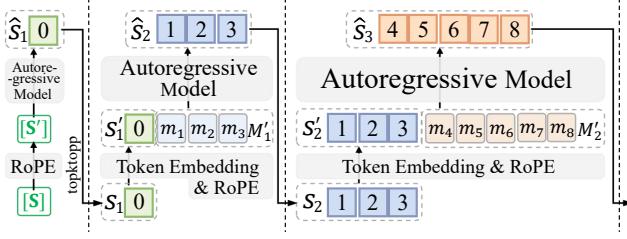
4

Figure 6. Inference process of EAR.

3.3.2. KV-Cache Design for Inference Acceleration

To further accelerate our inference speed, we design a KV cache mechanism compatible with the next-any-tokens prediction paradigm. As shown in Figure 6, our autoregressive generation process begins with a start token $[S]$ derived from the class embedding. After passing through the trained model, it generates the first token 0, and the corresponding key-value (KV) pairs of $[S]$ are stored in the KV cache.

In the second step, three mask tokens m_1, m_2, and m_3 are appended to the generated GT token 0 and input into the model together. These mask tokens are transformed into GT tokens 1, 2, and 3 respectively, and the KV pairs of tokens $0, m_1, m_2$, and m_3 are stored in the KV cache.

In the third step, five new mask tokens $m_4, m_5, m_6, m_7,$ and m_8 are appended to the previously generated GT tokens 1, 2, and 3, and passed through the model. These mask tokens are transformed into GT tokens 4 through 8. Importantly, the KV pairs of the newly generated GT tokens 1, 2, and 3 overwrite the cached KV pairs of the previous mask tokens m_1, m_2, and m_3, while the KV pairs of m_4 through m_8 are appended to the KV cache.

Compared to traditional autoregressive models such as LlamaGen, our method maintains the same sequence length during inference and introduces no additional computational cost.

By iterating this process, we effectively avoid redundant computation of previously generated tokens' key-value pairs, thereby accelerating the image generation during inference.

Algorithm 1 Next-any-tokens Prediction Train

1: **Input:** latent image gt, class token sos
2: **HyperParam:** steps K, mask token mt, mask M
3: $f = ARmodel(*), gt = \text{spiral}(gt)$
4: $MT = mt.\text{expand}(\text{len}(gt))$
5: $L_0 = \text{concat}(sos, gt, sos, MT)$
6: **for** $k = 1, \cdots, K$ **do**
7: $\quad L_k = f(L_{k-1}) \leftarrow M$
8: **end for**
9: $logits \leftarrow$ extract MT positions from L_K
10: $loss = \text{CrossEntropy}(logits, gt)$
11: **return** $loss$

Algorithm 2 Next-any-tokens Prediction Inference

1: **Input:** class token sos
2: **HyperParam:** steps K, tokens per step $(l_k)_{k=1}^{K}$, mask token mt, KV cache C
3: $f = ARmodel(concat(*, *)) \rightarrow (*, *)$
4: $L_0 = sos, L = [\,]$
5: **for** $k = 1, \cdots, K$ **do**
6: $\quad MT_k = mt.\text{expand}(l_k)$
7: $\quad \_, L_k = f(L_{k-1}, MT_k) \ and \ (L_{k-1}, MT_k) \rightarrow C$
8: $\quad L = \text{queue\_push}(L, L_k)$
9: **end for**
10: $L = \text{unspiral}(L)$
11: **return** generated latent image L

---

4. Experiments

4.1. Experimental Setup

To validate the effectiveness and scalability of the proposed EAR, we adopt a decoder-only Transformer architecture modified from the LlamaGen [31] implementation, and follow VAR [32] designs by integrating AdaLN-based conditioning into the transformer blocks. All ablation experiments, however, are performed without AdaLN to ensure a clean comparison of architectural factors. By introducing spiral unfolding and the next-any-tokens prediction mechanism, EAR is capable of generating images from the center outward in significantly fewer steps.

Class-conditional image generation. We evaluate EAR on the widely used ImageNet 256×256 dataset. Images are tokenized using the pre-trained image tokenizers proposed by LlamaGen [31] and XQGAN [17], both with a downsampling factor of 16. We experiment with three generation steps: 10, 16, and 31. Specifically, the 16-step and 31-step settings correspond to per-step token counts of $[1, 3, 5, 7, \ldots, 31]$ (where the number of tokens at step k is $2k - 1$) and $[1, 1, 2, 2, \ldots, 15, 15, 16]$ (where the number of tokens at step k is $(k + 1)//2$), respectively. These arrays correspond to $(l_k)_{k=1}^{K}$ in the pseudocode. Two configurations of mask tokens are used: one where a unified mask token is learned via a trainable embedding, and another where class tokens derived from class embeddings are used. All models are trained for 300 epochs with an initial learning rate of 2×10^{-4} using a step-wise learning rate scheduler. Inception Score (IS) and Frechet Inception Distance (FID) are used as evaluation metrics, computed by sampling 50,000 images using the official TensorFlow evaluation toolkit provided by ADM [7].

4.2. Main Results

We compare our proposed EAR models with recent state-of-the-art autoregressive image generation methods, including

5

Type	Model	FID ↓	IS ↑	Pre ↑	Rec ↑	#Para	#Step	Time ↓	GFLOPs
Diff.	ADM [7]	10.94	101.0	0.69	0.63	554M	250	265.75	–
	CDM [13]	4.88	158.7	–	–	–	8100	–	–
	LDM-4-G [27]	3.60	247.7	–	–	400M	250	–	–
	DiT-L/2 [24]	5.02	167.2	0.75	0.57	458M	250	38.12	–
	DiT-XL/2 [24]	2.27	278.2	0.78	0.62	675M	250	55.71	–
	L-DiT-3B *(upgraded from DiT [24])*	2.10	304.4	0.82	0.66	3.0B	250	>55.71	–
	L-DiT-7B *(upgraded from DiT [24])*	1.86	316.2	0.83	0.67	7.0B	250	>55.71	–
VAR	VAR-d16 [32]	3.30	274.4	0.84	0.51	310M	10	0.49	214.74
	VAR-d20 [32]	2.57	302.6	0.83	0.56	600M	10	0.50	268.42
	VAR-d30 [32]	1.92	323.1	0.82	0.63	2.0B	10	1.24	402.64
AR	LlamaGen-B [31]	5.46	193.6	0.84	0.46	111M	256	2.92	25.97
	LlamaGen-L [31]	3.80	248.3	0.83	0.52	343M	256	5.47	93.42
	LlamaGen-XL [31]	3.39	227.1	0.81	0.54	775M	256	8.08	220.46
AR	NAR-B [12]	4.65	212.3	0.83	0.47	130M	31	0.50	–
	NAR-M [12]	3.27	257.5	0.82	0.47	290M	31	0.71	–
	NAR-L [12]	3.06	263.9	0.81	0.53	372M	31	0.83	–
	NAR-XL [12]	2.70	277.5	0.81	0.58	816M	31	1.17	–
AR	PAR-B-4X-2.19rFid* [39]	6.21	204.4	0.86	0.46	111M	67	0.77	25.98
	PAR-L-4X-2.19rFid* [39]	4.32	189.4	0.87	0.43	343M	67	1.47	93.44
	PAR-XL-4X-2.19rFid* [39]	3.50	234.4	0.84	0.49	775M	67	2.16	220.50
AR (Ours)	EAR-B	4.64	218.7	0.82	0.48	98M	31	0.36	26.00
	EAR-L	3.06	261.4	0.83	0.54	326M	31	0.69	93.48
	EAR-XL	2.75	275.1	0.83	0.56	754M	31	1.03	220.58
AR (Ours)	EAR-B (adaLN)	4.14	237.7	0.83	0.50	140M	31	0.44	28.69
	EAR-L (adaLN)	2.76	266.4	0.83	0.57	477M	31	0.85	103.10
	EAR-XL (adaLN)	2.54	262.7	0.83	0.57	1.1B	31	1.37	243.14

Table 1. Quantitative comparison of various generative models on ImageNet 256×256. ∗: results implemented by NAR [12] using a 16 × 16 tokenizer. To mitigate the difference between tokenizers, FLOPs is calculated only for transformer blocks for generation of one image.
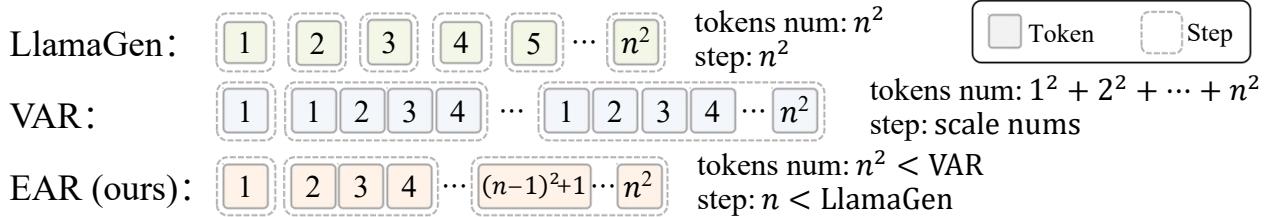

Figure 7. Comparisons of numbers of tokens at multiple generation steps among different methods.

LlamaGen [31], PAR [39], NAR [12], and scalable VAR models [32], as summarized in Table 5.

Our EAR framework achieves a compelling balance between generation quality and computational efficiency. EAR-XL attains a strong FID of 2.54 with only 31 autoregressive steps, requiring just 1.37 seconds per 32-image batch and 220 GFLOPs. This represents a significant improvement over prior autoregressive approaches. For instance, LlamaGen-XL [31] achieves an FID of 3.39 but requires 256 steps and 8.08 seconds per batch, highlighting EAR's superior step efficiency and faster inference (83.0% speedup) despite comparable parameter counts. Overall, our EAR models deliver better FID and IS scores at compa-

rable parameter scales and inference speeds relative to the NAR [12] counterparts. For instance, EAR-B achieves a lower FID (4.14 vs. 4.65) than NAR-B while using a similar number of parameters (140M vs. 130M) and attaining a higher IS (237.7 vs. 212.3). Notably, when compared with VAR [32] models under a similar parameter scale, our EAR-L achieves a lower FID (2.76 vs. 3.30) and requires significantly fewer GFLOPs (103.10G vs. 214.74G).

These results demonstrate EAR's ability to deliver competitive or superior image quality while drastically reducing computational overhead, establishing a new efficiency benchmark for autoregressive visual generation. The synergy between spiral token unfolding and parallel next-any-
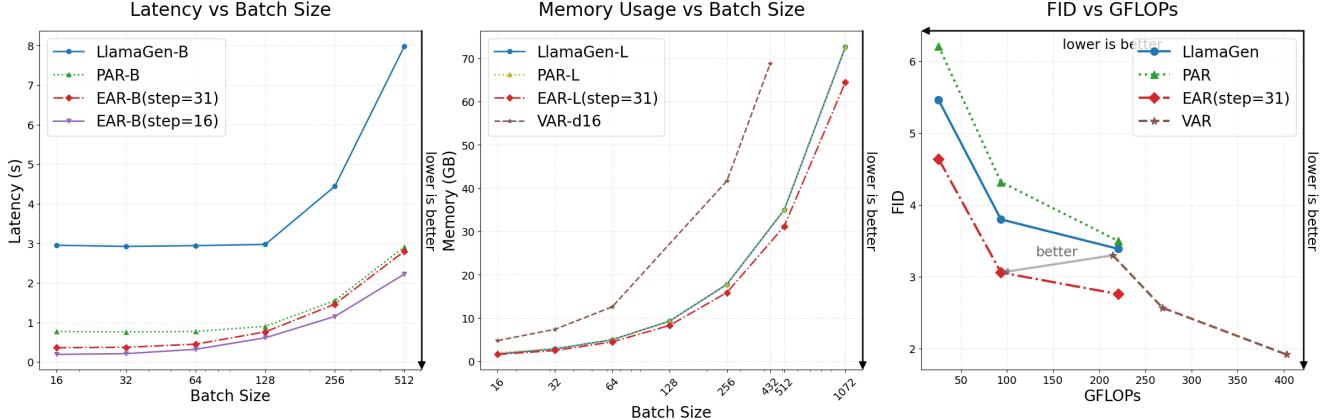
Figure 8. Efficiency comparisons between vanilla AR, PAR, VAR and the proposed EAR (without adaLN).

tokens prediction enables high-fidelity synthesis without sacrificing scalability, addressing a core limitation of traditional autoregressive approaches.

4.3. Comparison of Model Inference

As shown in Figure 7, suppose the token size of a single image is $n \times n$. For LlamaGen, it needs to predict $n \times n$ tokens, and due to the next-token prediction paradigm, the number of inference steps required to generate an image is also $n \times n$.

For VAR [32], which follows the next-scale prediction paradigm, it needs to predict the sum of tokens across all scales. The number of inference steps corresponds to the number of scales used in the model, which is 10 as reported in the paper.

In contrast, our EAR model also only needs to predict $n \times n$ tokens, which is significantly fewer than the total number of multi-scale tokens used in VAR [32]. Leveraging our proposed next-any-tokens prediction, we can generate an image in an arbitrary number of steps. To balance generation quality and speed, we empirically set the number of steps to n, which is significantly fewer than the $n \times n$ steps required by LlamaGen [31]. As illustrated in Figure 8, our inference time is significantly lower than that of LlamaGen [31], and as the batch size increases, our memory usage remains far below that of VAR [32]. Meanwhile, EAR achieves an excellent trade-off between FID and GFLOPs, demonstrating a well-balanced combination of generation quality and computational efficiency.

4.4. Ablation Study

4.4.1. Effect of Step Numbers

To investigate the relationship between decoding steps and generation quality, we conduct experiments using LlamaGen's tokenizer with 10 and 16 steps. As shown in Table 2, using 16 steps significantly improves image quality

compared to 10 steps. Table 3 also indicates that 31 steps achieve better results than 16 steps.

Model	Params	FID	IS	Steps	Time (s)
EAR-B	111M	7.13	217.9	10	0.12
EAR-B	111M	6.39	231.8	16	0.21
EAR-L	343M	4.69	272.1	16	0.51
EAR-XL	775M	3.90	285.0	16	0.88

Table 2. Effect of different step numbers (with LlamGen's VQ-VAE [31]).

Model	Params	FID	IS	Steps	Time (s)
EAR-B	98M	5.49	225.2	16	0.21
EAR-L	326M	3.55	256.44	16	0.51
EAR-XL	754M	2.92	266.03	16	0.88
EAR-B	98M	4.64	218.7	31	0.36
EAR-L	326M	3.06	261.4	31	0.69
EAR-XL	754M	2.75	275.1	31	1.03

Table 3. Effect of different step numbers (with XQGAN's VQ-VAE [17]).

4.4.2. Effect of Mask Token Selection

In the experimental stage, we explored two types of mask tokens: one derived from class embeddings as the category-specific start token, and the other being a unified, learnable mask embedding shared across all categories. As shown in the Table 4, the results show that, under the same conditions, using the unified mask token achieves better performance than using the class embedding-derived mask tokens.

4.5. Image Extension Task

Thanks to our spiral token unfolding strategy, our EAR model can flexibly handle a variety of image extension tasks under different input conditions. In this section, we demonstrate its capability across four settings.

7

Model	Params	FID	IS	Steps
Using Class Mask Token				
EAR-B	111M	6.78	209.3	16
EAR-L	343M	4.88	238.8	16
EAR-XL	775M	4.03	238.4	16
Using Unified Mask Token				
EAR-B	111M	6.39	231.8	16
EAR-L	343M	4.69	272.1	16
EAR-XL	775M	3.90	285.0	16

Table 4. Comparison of class-specific and unified mask tokens. Results indicate that unified mask tokens yield better performance across all model sizes.
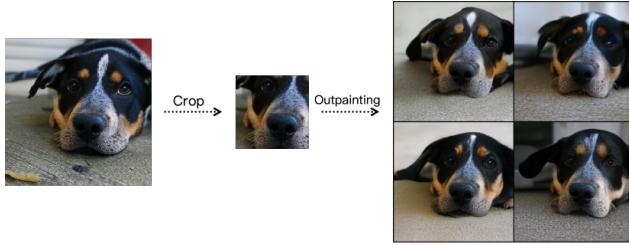
4.5.1. Extension from Identical Center Tokens

Figure 9. Image extension results based on the same center 8×8 tokens.

We select a 256×256 image and tokenize it into a 16×16 grid using a VQ-VAE encoder. The central 8×8 tokens, obtained via our spiral token unfolding strategy, are used as the condition input to our EAR model. The model then autoregressively predicts the full 16×16 token map, which is decoded back into the image domain via VQ-VAE. As shown in Figure 9, the generated images effectively reconstruct diverse contents based on the same center region.
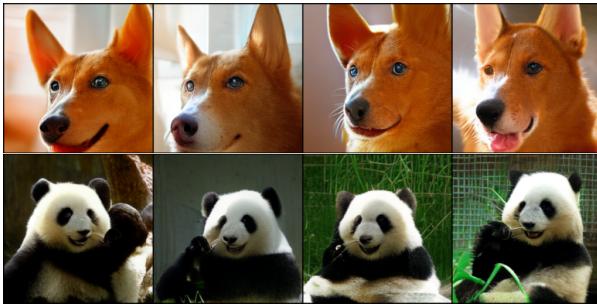
4.5.2. Extension from Different Crops

Figure 10. Results of image extension task. The generation depends on the given images (128×128) of the central region.

We apply different cropping and flipping operations on the same ImageNet image to produce multiple 256×256

variants. Following the same extension procedure as above, our model generates diverse completions reflecting variations in the input crops, as illustrated in Figure 10.

4.5.3. Extension from Scaled-down Full Images

Figure 11. Image extension from downscaled inputs.

Given an original 256×256 image, we first downscale it to lower resolutions, specifically 192×192 and 128×128. These resized images are then tokenized by the VQ-VAE encoder into 12×12 and 8×8 token grids, respectively. We treat these token grids as partial observations and use them as conditional inputs to our EAR model, which autoregressively predicts the full 16×16 token sequence. Finally, the generated token map is decoded back into the image domain using the VQ-VAE decoder, resulting in a complete 256×256 image, as illustrated in Figure 11.
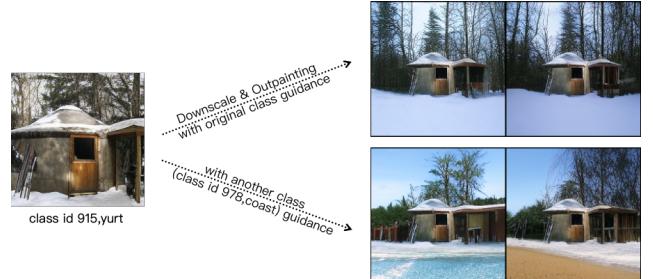
4.5.4. Cross-Class Image Extension

Figure 12. Image extension from downscaled inputs.

We conducted an interesting cross-class image extension experiment. A 256×256 image was downscaled and tokenized with VQ-VAE, and token generation was guided by an embedding from another class. As shown in Figure 12, the original image depicts a yurt in a snowy environment. With the original class embedding, the generated result preserves the snow scene; with the "coast" embedding, the yurt unexpectedly appears on a beach, revealing an intriguing cross-semantic transformation.

5. Conclusion

In this paper, we address a key challenge in visual autoregressive modeling: achieving high-fidelity next-token prediction while remaining efficient for high-resolution image synthesis. Inspired by human visual cognition, we propose Expanding Autoregressive Modeling (EAR),

which integrates a spiral token unfolding strategy with next-any-token prediction. This center-outward process preserves spatial continuity and enables controlled multi-token parallelization. On ImageNet 256×256, EAR surpasses prior autoregressive baselines, generating images substantially faster than LlamaGen [31] and matching or exceeding VAR [32] in quality at lower computational cost. These results show that EAR offers an efficient, cognitively inspired alternative to conventional multi-scale generation.

References

[1] Sotiris Anagnostidis, Dario Pavllo, Luca Biggio, Lorenzo Noci, Aurelien Lucchi, and Thomas Hofmann. Dynamic context pruning for efficient and interpretable autoregressive transformers. *Advances in Neural Information Processing Systems*, 36:65202–65223, 2023. 3

[2] C. N. Boehler, J. K. Tsotsos, M. A. Schoenfeld, H.-J. Heinze, and J.-M. Hopf. The center-surround profile of the focus of attention arises from recurrent processing in visual cortex. *Cerebral Cortex*, 19(4):982–991, 2009. Epub 2008 Aug 28. 2, 3

[3] 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. 2

[4] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020. 2

[5] Xi Chen, Nikhil Mishra, Mostafa Rohaninejad, and Pieter Abbeel. Pixelsnail: An improved autoregressive generative model. In *International conference on machine learning*, pages 864–872. PMLR, 2018. 2

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019. 2

[7] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. 5, 6

[8] 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. 2

[9] 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. 2

[10] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1

[11] Yefei He, Feng Chen, Yuanyu He, Shaoxuan He, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. Zipar: Accelerating auto-regressive image generation through spatial locality. *arXiv preprint arXiv:2412.04062*, 2024. 3

[12] Yefei He, Yuanyu He, Shaoxuan He, Feng Chen, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. Neighboring autoregressive modeling for efficient visual generation, 2025. 6

[13] 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. 6

[14] 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. 2

[15] Haopeng Li, Jinyue Yang, Guoqi Li, and Huan Wang. Autoregressive image generation with randomized parallel decoding, 2025. 1

[16] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37:56424–56445, 2024. 3

[17] Xiang Li, Kai Qiu, Hao Chen, Jason Kuen, Jiuxiang Gu, Jindong Wang, Zhe Lin, and Bhiksha Raj. Xq-gan: An open-source image tokenization framework for autoregressive generation, 2024. 5, 7

[18] Marcel Linka, Harun Karimpur, and Benjamin de Haas. Protracted development of gaze behaviour. *Nature Human Behaviour*, 2025. 2, 3

[19] Enshu Liu, Xuefei Ning, Yu Wang, and Zinan Lin. Distilled decoding 1: One-step sampling of image autoregressive models with flow matching. *arXiv preprint arXiv:2412.17153*, 2024. 3

[20] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 2

[21] Jinming Lou, Wenyang Luo, Yufan Liu, Bing Li, Xinmiao Ding, Weiming Hu, Jiajiong Cao, Yuming Li, and Chenguang Ma. Token caching for diffusion transformer acceleration. *arXiv preprint arXiv:2409.18523*, 2024. 2

[22] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720, 2023. 2

[23] Wael Mattar, Idan Levy, Nir Sharon, and Shai Dekel. Wavelets are all you need for autoregressive image generation. *arXiv preprint arXiv:2406.19997*, 2024. 2

[24] 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. 2, 6

[25] 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. 2

[26] Scott Reed, Aäron van den Oord, Nal Kalchbrenner, Victor Bapst, Matt Botvinick, and Nando De Freitas. Generating interpretable images with controllable structure. 2016. 2

[27] 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. 2, 6

[28] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017. 2

[29] Michael Santacroce, Zixin Wen, Yelong Shen, and Yuanzhi Li. What matters in the structured pruning of generative language models? *arXiv preprint arXiv:2302.03773*, 2023. 2

[30] Xuan Shen, Zhao Song, Yufa Zhou, Bo Chen, Jing Liu, Ruiyi Zhang, Ryan A Rossi, Hao Tan, Tong Yu, Xiang Chen, et al. Numerical pruning for efficient autoregressive models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 20418–20426, 2025. 3

[31] 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. 2, 5, 6, 7, 9

[32] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction, 2024. 5, 6, 7, 9

[33] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021. 2

[34] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016. 2

[35] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 1, 2

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[37] Zhongwei Wan, Xin Wang, Che Liu, Samiul Alam, Yu Zheng, Jiachen Liu, Zhongnan Qu, Shen Yan, Yi Zhu, Quanlu Zhang, et al. Efficient large language models: A survey. *arXiv preprint arXiv:2312.03863*, 2023. 1

[38] Junke Wang, Yi Jiang, Zehuan Yuan, Bingyue Peng, Zuxuan Wu, and Yu-Gang Jiang. Omnitokenizer: A joint image-video tokenizer for visual generation. *Advances in Neural Information Processing Systems*, 37:28281–28295, 2024. 2

[39] Yuqing Wang, Shuhuai Ren, Zhijie Lin, Yujin Han, Haoyuan Guo, Zhenheng Yang, Difan Zou, Jiashi Feng, and Xihui Liu. Parallelized autoregressive visual generation, 2025. 1, 6

[40] Lu Yu and Wei Xiang. X-pruner: explainable pruning for vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24355–24363, 2023. 2

[41] 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. 2

[42] Lei Zhu, Fangyun Wei, Yanye Lu, and Dong Chen. Scaling the codebook size of vq-gan to 100,000 with a utilization rate of 99%. *Advances in Neural Information Processing Systems*, 37:12612–12635, 2024. 2

[43] Mingjian Zhu, Yehui Tang, and Kai Han. Vision transformer pruning. *arXiv preprint arXiv:2104.08500*, 2021. 2

Model	FID ↓	IS ↑	#Para	#Step	Time
EAR-B	4.64	218.7	98M	31	0.36
EAR-L	3.06	261.4	326M	31	0.69
EAR-XL	2.75	275.1	754M	31	1.03
EAR-XXXL	2.24	271.1	3.85B	31	1.77
EAR-B	5.49	225.2	98M	16	0.51
EAR-L	3.55	256.44	326M	16	0.69
EAR-XL	2.92	266.03	754M	16	0.81
EAR-B (adaLN)	4.14	237.7	140M	31	0.44
EAR-L (adaLN)	2.76	266.4	477M	31	0.85
EAR-XL (adaLN)	2.54	262.7	1.1B	31	1.37

Table 5. Quantitative comparison of EAR models and their AdaLN variants on ImageNet 256×256. Lower FID and higher IS indicate better generative performance.

A. Overall Quantitative Comparison

Table 5 and Figure 13 14 present a quantitative comparison of our proposed EAR models and their AdaLN variants on ImageNet 256×256. The metrics include **FID** (Fréchet Inception Distance, lower is better) and **IS** (Inception Score, higher is better), along with the number of parameters (#Para), generation steps (#Step), and time for 32-image batch (in seconds).

From the results, several observations can be made:

- **Scaling Effect:** Increasing model size from EAR-B to EAR-XXXL consistently reduces FID and increases IS, indicating better image quality and more realistic samples.
- **Sampling Steps:** Reducing the number of generation steps (from 31 to 16) slightly increases FID but decreases inference time, demonstrating a trade-off between quality and speed.
- **AdaLN Variants:** Incorporating AdaLN generally improves IS while maintaining or slightly reducing FID compared to the baseline models with the same architecture, showing the effectiveness of adaptive normalization in enhancing class-conditional generation.
- **Computational Efficiency:** Even for the largest model (EAR-XL(adaLN)), the per-image generation time remains reasonable (1.37 s) with a single generation step, illustrating that our approach scales efficiently.

In summary, the table demonstrates that EAR models can generate high-quality images with controllable speed-accuracy trade-offs, and adaptive normalization further enhances class-conditional generation performance.
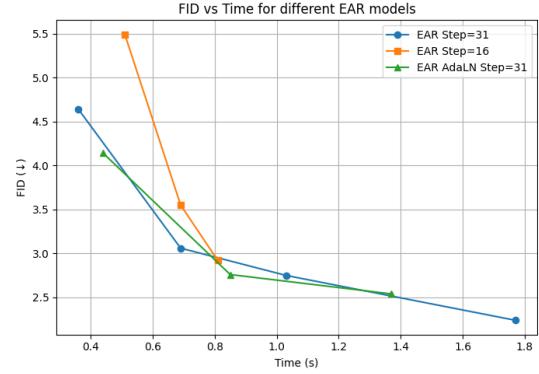

Figure 13. FID comparison with respect to inference time across different EAR model variants. Models with more parameters generally achieve lower FID but require longer inference time. The AdaLN-enhanced models consistently outperform their vanilla counterparts at similar time costs.
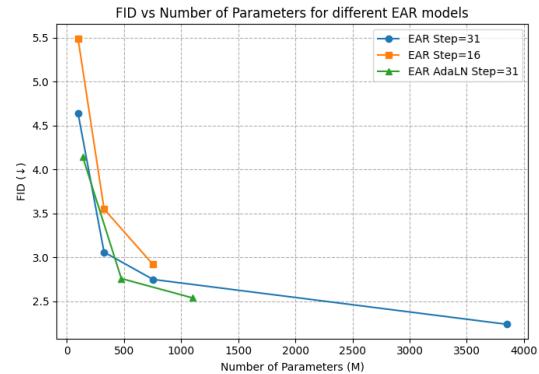

Figure 14. FID comparison with respect to model parameter size. Larger EAR models achieve better generative quality with steadily decreasing FID. The AdaLN variants show improved efficiency by obtaining lower FID with fewer parameters.

B. Details of Mask Tokens

To enable parallel generation, we adopt two types of mask tokens in our framework. Although both serve similar purposes, their implementations differ in practice. We provide detailed explanations below.

B.1. Class Mask Token

The class token is directly obtained from a class embedding and naturally serves as a semantic indicator for the target category. It inherently functions as the start token for generation, making it unnecessary to introduce a separate start token. Therefore, the causal masking strategy in the Trans-

former remains unchanged, as described in the main text.

B.2. Unified Mask Token

In contrast, the unified mask token is an additional learnable embedding that does not contain class information. To ensure consistent performance, we slightly modify the model structure (as shown in the main figure of the paper): the class token derived from the class embedding is prepended to both the ground-truth token sequence and the mask token sequence. This ensures that the mask token always has access to class-level information, allowing it to guide class-conditional generation.

Importantly, this modification does not alter the overall model design. The task can still be treated as image generation with $(n \times n + 1)$ tokens, and the additional token has negligible impact on computational efficiency.

C. Next-any-tokens Prediction Method

In this section, we explain the training and inference procedures of the proposed *Next-any-tokens Prediction* framework, which is designed for efficient autoregressive latent image generation. Our method generalizes traditional next-token prediction by allowing multiple positions (or "any tokens") to be predicted in parallel at each step.

C.1. Training Procedure

Algorithm 3 Next-any-tokens Prediction Train

1: **Input:** latent image gt, class token sos
2: **HyperParam:** steps K, mask token mt, mask M
3: $f = ARmodel(*), gt = \text{spiral}(gt)$
4: $MT = mt.\text{expand}(\text{len}(gt))$
5: $L_0 = \text{concat}(sos, gt, sos, MT)$
6: **for** $k = 1, \cdots, K$ **do**
7: $\quad L_k = f(L_{k-1}) \leftarrow M$
8: **end for**
9: $logits \leftarrow$ extract MT positions from L_K
10: $loss = \text{CrossEntropy}(logits, gt)$
11: **return** $loss$

Algorithm 3 shows the training procedure. The input to the model consists of a latent image gt and a class token sos. The latent image is first reordered using a spiral scan (`spiral(gt)`), which ensures that the autoregressive model generates tokens from the center outwards, mimicking human visual perception.

A mask token mt is expanded to match the length of the latent sequence, forming a masked sequence L_0 concatenated with sos, the ground truth latent tokens gt, another sos, and MT (the mask token sequence). This sequence allows the model to learn to predict masked positions while conditioning on known tokens and class information.

The model f (an autoregressive transformer) is applied iteratively for K steps. At each step, the sequence L_{k-1} is updated according to the mask M, gradually refining the prediction of masked positions. After K iterations, the logits corresponding to masked token positions are extracted and used to compute a cross-entropy loss against the ground truth latent image. This loss is then used to optimize the model parameters.

Formally:

$$L_k = f(L_{k-1}) \leftarrow M, \quad k = 1, \ldots, K$$

$$\text{loss} = \text{CrossEntropy}(\text{logits at MT positions}, gt)$$

C.2. Inference Procedure

Algorithm 4 Next-any-tokens Prediction Inference

1: **Input:** class token sos
2: **HyperParam:** steps K, tokens per step $(l_k)_{k=1}^{K}$, mask token mt, KV cache C
3: $f = ARmodel(concat(*, *)) \rightarrow (*, *)$
4: $L_0 = sos, L = [\,]$
5: **for** $k = 1, \cdots, K$ **do**
6: $\quad MT_k = mt.\text{expand}(l_k)$
7: $\quad \_, L_k = f(L_{k-1}, MT_k) \text{ and } (L_{k-1}, MT_k) \rightarrow C$
8: $\quad L = \text{queue\_push}(L, L_k)$
9: **end for**
10: $L = \text{unspiral}(L)$
11: **return** generated latent image L

Algorithm 4 illustrates the inference process. Starting from a class token sos, we generate a latent image in K steps. At each step k, a mask token sequence MT_k of length l_k is concatenated with the previously generated sequence L_{k-1}. The autoregressive model f predicts the next set of tokens, and the output L_k is stored in a queue L for accumulation. A key-value (KV) cache C is maintained to efficiently store transformer attention states across steps, reducing redundant computation.

After completing all K steps, the accumulated sequence L is reordered back to the original spatial layout using `unspiral(L)`, producing the final latent image.

The inference procedure can be summarized as:

$$L = \text{queue\_push}(L, f(L_{k-1}, MT_k)), \quad k = 1, \ldots, K$$

$$\text{output image} = \text{unspiral}(L)$$

D. Visualizations of Generation Results

We visualize the generation results of our EAR on ImageNet 256×256 images. As shown in Figure 15–18, the generated images exhibit high visual fidelity with crisp details, coherent object boundaries, and semantically consistent layouts.

Class ID: 2

Class ID: 10

Class ID: 20

Class ID: 22

Class ID: 24

Class ID: 33

Figure 15. Generated images by EAR-XL with AdaLN (Page 1 of 4).
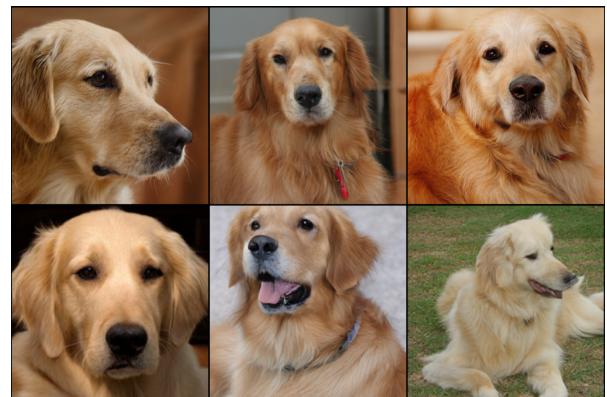
Class ID: 88

Class ID: 105

Class ID: 108

Class ID: 207

Class ID: 248

Class ID: 258

Figure 16. Generated images by EAR-XL with AdaLN (Page 2 of 4).

Class ID: 360

Class ID: 387

Class ID: 415

Class ID: 466

Class ID: 483

Class ID: 780

Figure 17. Generated images by EAR-XL with AdaLN (Page 3 of 4).

Class ID: 928

Class ID: 933

Class ID: 972

Class ID: 973

Class ID: 980

Class ID: 985

Figure 18. Generated images by EAR-XL with AdaLN (Page 4 of 4).