Randomized Autoregressive Visual Generation

Qihang Yu Ju He Xueqing Deng Xiaohui Shen Liang-Chieh Chen
ByteDance

Abstract

*This paper presents **R**andomized **A**uto**R**egressive modeling (**RAR**) for visual generation, which sets a new state-of-the-art performance on the image generation task while maintaining full compatibility with language modeling frameworks. The proposed RAR is simple: during a standard autoregressive training process with a next-token prediction objective, the input sequence—typically ordered in raster form—is randomly permuted into different factorization orders with a probability r, where r starts at 1 and linearly decays to 0 over the course of training. This annealing training strategy enables the model to learn to maximize the expected likelihood over all factorization orders and thus effectively improve the model's capability of modeling bidirectional contexts. Importantly, RAR preserves the integrity of the autoregressive modeling framework, ensuring full compatibility with language modeling while significantly improving performance in image generation. On the ImageNet-256 benchmark, RAR achieves an FID score of **1.48**, not only surpassing prior state-of-the-art autoregressive image generators but also outperforming leading diffusion-based and masked transformer-based methods. Code and models are available at* https://github.com/bytedance/1d-tokenizer.

1. Introduction

AutoRegressive (AR) models have driven remarkable advancements across both natural language processing and computer vision tasks in recent years. In language modeling, they serve as the fundamental framework for Large Language Models (LLMs) such as GPT [45], Llama [64, 65], and Gemini [62], along with other state-of-the-art models [1, 72]. In the realm of computer vision, autoregressive models[1] have also shown substantial potential, delivering competitive performance in image generation tasks [22, 37, 41, 52, 53, 57, 75, 76] to diffusion models [6, 18, 29, 38, 47, 55] or non-autoregressive transformers [10, 34, 70, 77–79]. More importantly, autoregressive modeling is emerging as a promising pathway toward unified models across multiple modalities and tasks [5, 9, 14, 60, 61, 71].

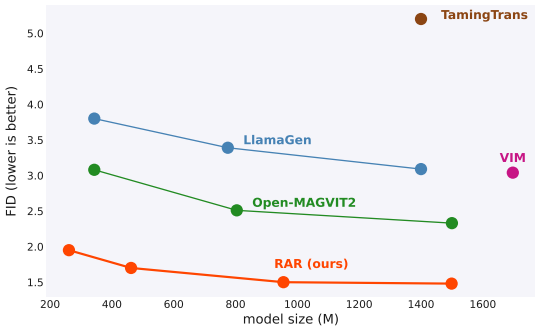


Figure 1. **Comparison among different language modeling compatible autoregressive (AR) image generators.** The proposed RAR demonstrates significant improvements over previous AR methods. RAR-B, with only *261M* parameters, achieves an FID score of 1.95, outperforming both LlamaGen-XXL (1.4B parameters) and Open-MAGVIT2-XL (1.5B parameters).

Despite the dominance of autoregressive models in language modeling, they often yield suboptimal performance in comparison to diffusion models or non-autoregressive transformers in visual generation tasks [41, 57]. This discrepancy can be attributed to the inherent differences between text and visual signals. Text is highly compact and semantically meaningful, while visual data tends to be more low-level and redundant [30, 79], making bidirectional context modeling more critical. For instance, several studies [7, 21, 38] have demonstrated that causal attention applied to image tokens leads to inferior performance compared to bidirectional attention in vision tasks.

To address this, recent works [38, 63] have attempted to reintroduce bidirectional attention by redesigning the autoregressive formulation, achieving state-of-the-art results in image generation. However, these approaches often deviate from the traditional autoregressive paradigm. For example,

[1]While MaskGIT-style models [10] could be classified as "generalized autoregressive models" as defined in [38], in this paper, we primarily use the term "autoregressive" to refer to GPT-style models [22, 57, 75], which are characterized by ***causal*** attention, ***next-token*** prediction, and operate ***without*** the need for mask tokens as placeholders.

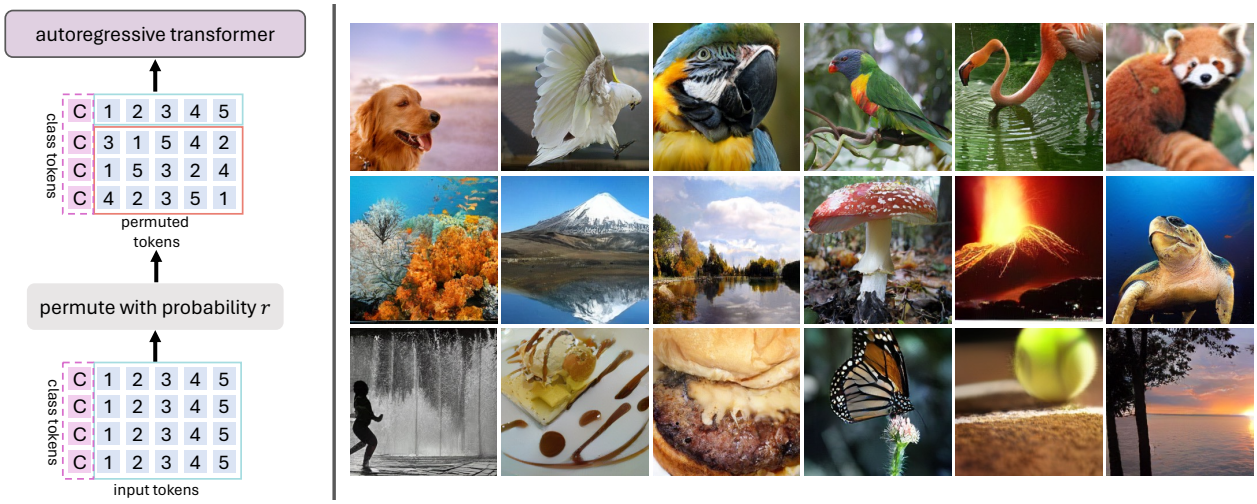


Figure 2. **Overview of the proposed Randomized AutoRegressive (RAR) model, which is fully compatible with language modeling frameworks.** *Left*: RAR introduces a randomness annealing training strategy to enhance the model's ability to learn bidirectional contexts. During training, the input sequence is randomly permuted with a probability r, which starts at 1 (fully random permutations) and linearly decreases to 0, transitioning the model to a fixed scan order, such as raster scan, by the end of training. *Right*: Randomly selected images generated by RAR, trained on ImageNet.

VAR [63] shifts from next-token prediction to next-scale prediction, enabling bidirectional attention within each scale, and MAR [38] generalizes MaskGIT-style framework [10] to the autoregressive definition, which naturally introduces back the bidirectional attention. While effective, these modifications complicate their integration into universal transformer architectures that aim to unify different modalities, which proves to work well with conventional autoregressive modeling [60, 61].

In this paper, we aim to enhance the generation quality of autoregressive image models while preserving the core autoregressive structure, maintaining compatibility with language modeling frameworks. Specifically, we enable bidirectional context learning within an autoregressive transformer by maximizing the expected likelihood over all possible factorization order. In this way, all tokens will be trained and predicted under all possible contexts, facilitating learning bidirectional representation. Moreover, we introduce a permutation probability r, which controls the ratio of training data between a random factorization order and the standard raster order. Initially, r is set to 1 (fully random factorization) and it linearly decays to 0 over the course of training, gradually reverting the model to the raster order commonly used by other autoregressive image generators.

To this end, we present a simple, effective, and scalable autoregressive model training paradigm named **R**andomized **A**uto**R**egressive modeling (**RAR**). RAR retains the original autoregressive model architecture and formulation, ensuring full compatibility with language modeling. At the same time, it significantly improves the generation quality of autoregressive models at no additional cost. On the ImageNet-256 benchmark [16], RAR achieves an FID score of 1.48, substantially outperforming previous state-of-the-art autoregressive image generators, as illustrated in Fig. 1. By addressing the limitations of unidirectional context modeling, RAR represents a critical step towards autoregressive visual generation and opens up new possibilities for further advancements in the field.

2. Related Work

Autoregressive Language Modeling. The advent of autoregressive language models [1–4, 9, 13, 20, 45, 48, 49, 62, 64, 65, 72] has paved a promising path toward general-purpose AI systems. At the core of these models is a simple yet powerful next-token prediction paradigm, where the objective is to predict the next word or token in a sequence based on preceding inputs. This approach has demonstrated both scalability, as evidenced by scaling laws, and versatility through zero-shot generalization, enabling explorations beyond traditional language tasks to diverse modalities.

Autoregressive Visual Modeling. Pioneering research [12, 27, 46, 67, 68] in autoregressive visual modeling has focused on representing images as sequences of pixels. Nevertheless, inspired by advancements in autoregressive language modeling, a subsequent wave of studies has transitioned to modeling images as sequences of discrete-valued tokens [22, 50, 51, 69, 75], resulting in notable improvements in performance. This direction has been further explored through efforts [41, 57] aimed at enhancing tokenization quality and leveraging modern autoregressive architectures initially developed for language tasks. However, all of these

works strictly adhere to a raster-scan order for processing pixels or tokens, resulting in a unidirectional information flow that is sub-optimal for visual modeling. In this work, we instead explore learning across all possible factorization orders to enhance bidirectional context learning while retaining the core autoregressive framework.

Other Visual Generation Models. In addition to autoregressive visual modeling, there have been numerous efforts in exploring other formats of visual generation models, including generative adversarial networks (GANs) [8, 26, 33], diffusion models [18, 23, 32, 39, 47, 54, 73], masked transformers [10, 11, 70, 77, 79], scale-wise autoregressive modeling (VAR) [43, 59, 63, 81], and masked autoregressive modeling with diffusion loss (MAR) [24, 38]. It is worth noting that MAR [38] also experimented a random order based AR framework similar to the proposed RAR. However, as indicated in our experiments (see Sec. 4.2), simply replacing the raster order with random order only brings marginal improvement, coinciding the observation in [38]. This further demonstrates the importance on the randomness annealing strategy in RAR, leading to a substantial improvement for the AR image generators.

3. Method

In this section, we first provide an overview of autoregressive modeling in Sec. 3.1, followed by our proposed Randomized AutoRegressive modeling (RAR) in Sec. 3.2.

3.1. Background

We provide a brief overview of autoregressive modeling with a next-token prediction objective. Given a discrete token sequence $\mathbf{x} = [x_1, x_2, \cdots, x_T]$, the goal of autoregressive modeling is to maximize the likelihood of the sequence under a forward autoregressive factorization. Specifically, the objective is to maximize the joint probability of predicting the current token x_t based on all preceding tokens $[x_1, x_2, \cdots, x_{t-1}]$, $\forall t = 1, \cdots, T$:

$$\max_{\theta} \quad p_\theta(\mathbf{x}) = \prod_{t=1}^{T} p_\theta(x_t | x_1, x_2, \cdots, x_{t-1}), \qquad (1)$$

where p_θ denotes a token distribution predictor with a model parameterized by θ.

As shown in the equation, each token x_t at position t is conditioned solely on the preceding tokens, which limits context modeling to a unidirectional manner. This contrasts with methods such as masked transformer [10, 70, 77, 78] and diffusion models [32, 39, 47, 54], which can leverage bidirectional context at the training time. Additionally, while natural language has an inherent sequential order (left-to-right in most languages), image data lacks a clear, predefined order for processing tokens. Among the possible orders for image generation, the row-major order (*i.e.*, raster scan) is the most widely adopted and has demonstrated superior performance compared to other alternatives [22].

3.2. RAR: Randomized AutoRegressive Modeling

Visual signals inherently exhibit bidirectional correlations, making effective global context modeling essential. However, conventional autoregressive models rely on causal attention masking, which enforces a unidirectional dependency on the token sequence, contradicting the nature of visual data, as noted in prior works [7, 21, 38], where bidirectional attention works significantly better than causal attention for visual modality. Furthermore, there is no universally "correct" way to arrange image tokens into a causal sequence. While the widely adopted raster order has achieved some success, it introduces biases in the autoregressive training process. For instance, each token is conditioned solely on the preceding tokens in the scanning order, restricting the model's ability to learn dependencies from other directions.

To address these challenges, we propose a randomized autoregressive modeling approach that incorporates optimization objective with bidirectional context:

$$\max_{\theta} \quad p_\theta(\mathbf{x}) = \prod_{t=1}^{T} p_\theta(x_t | x_1, \cdots, x_{t-1}, x_{t+1}, \cdots, x_T). \qquad (2)$$

Unlike BERT-style [17] or MaskGIT-style [10] methods, our method follows the permuted objective approach [66, 74], where the model is trained in an autoregressive manner across all possible factorization orders. This enables the model to gather bidirectional context while preserving the autoregressive framework *in expectation*. Formally, we have:

$$\max_{\theta} \quad p_\theta(\mathbf{x}) = \mathbb{E}_{\tau \sim \mathcal{S}_T} \left[\prod_{t=1}^{T} p_\theta(x_{\tau_t} | x_{\tau_{<t}}) \right], \qquad (3)$$

where $\mathcal{S}_T$ denotes the set of all possible permutations of the index sequence $[1, 2, \cdots, T]$, and τ represents a randomly sampled permutation from $\mathcal{S}_T$. The notation τ_t refers to the t-th element in the permuted sequence, and $\tau_{<t}$ represents all preceding positions to τ_t. Since the model parameters θ are shared across all sampled factorization orders, each token x_t is exposed to every possible context and learns relationships with every other token x_i $\forall i \neq t$, during training. This allows the model to effectively capture bidirectional context while preserving the integrity of the autoregressive formulation.

Although simple, this modification significantly improves image generation performance, highlighting the power of bidirectional context in improving autoregressive image generator capability. Our findings align with those observed in autoregressive training for language modeling in NLP [9, 17, 66, 74] as well.

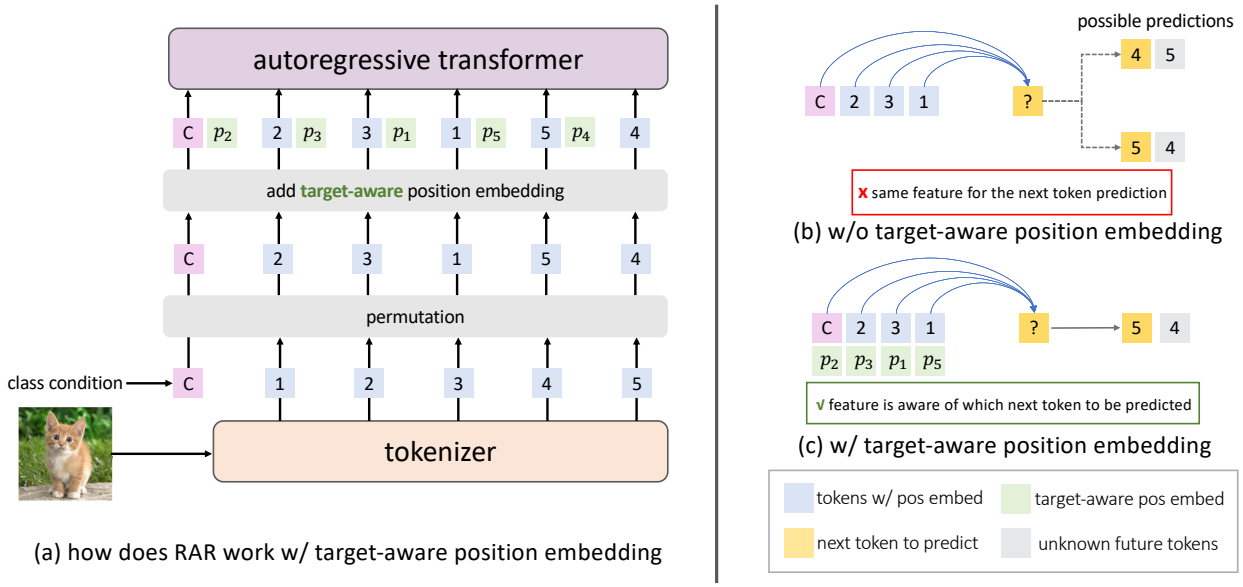


Figure 3. **Illustration of the target-aware positional embedding.** Subfigure (a) shows the training process of the proposed Randomized AutoRegressive (RAR) model, along with the target-aware position embedding. Following Vision Transformer [19], images are tokenized into patches with original position embeddings (blue tokens). The token sequence is then randomly permuted, with the target-aware positional embeddings (green tokens) added to guide the model. Subfigures (b) and (c) highlight the importance of the target-aware positional embedding: (b) demonstrates a failure case where both permuted sequences yield identical prediction logits, while (c) shows that the target-aware positional embedding correctly guides the model to predict the next token accurately.

Discussion. While the permutation objective allows for bidirectional context learning within the autoregressive framework *in expectation*, it remains challenging to fully capture "global context" during the generation process. This is because there are always some tokens generated before others, without having access to the full global context. This limitation is not unique to autoregressive methods [22, 57] but also present in non-autoregressive models [10]. Techniques such as resampling or refinement [28, 44] may help address this issue by ensuring that every token is generated with sufficient context. However, such designs may complicate the system; thus, exploring such solutions lies beyond the scope of this paper and is left for future work.

Target-aware Positional Embedding. One limitation of the permuted training objective is that standard positional embeddings may fail in certain scenarios. For instance, consider two different permutations: $\tau_a = [1, 2, \cdots, T-2, T-1, T]$ and $\tau_b = [1, 2, \cdots, T-2, T, T-1]$ (*i.e.*, only the last two tokens' positions are swapped). When predicting the second to last token, both permutations will yield identical features and thus identical prediction logits, even though they correspond to different ground-truth labels (*i.e.*, $p_\theta(x_{\tau_{T-1}}|x_{\tau_1}, x_{\tau_2}, \cdots, x_{\tau_{T-2}})$ is the same for both permutations τ_a and τ_b). This problem, in a general randomized autoregressive training process and beyond this specific example, can happen for all token locations except the last one (since the last token does not need to predict next token). To address this issue, we introduce an additional set of positional embeddings, which we refer to as *target-aware positional embeddings*. These embeddings encode information about which token is being predicted next.

Formally, we define a set of target-aware positional embeddings $\mathbf{p}_{ta} = [p_1, p_2, \cdots, p_T]$. The positional embedding corresponding to the next token is added to the current token embedding, resulting in a target-aware token embedding $\hat{\mathbf{x}}_\tau$:

$$\hat{\mathbf{x}}_\tau = \mathbf{x}_\tau + \mathbf{p}_\tau = [x_{\tau_1} + p_{\tau_2}, x_{\tau_2} + p_{\tau_3}, \cdots, x_{\tau_{T-1}} + p_{\tau_T}, x_{\tau_T}], \tag{4}$$

where $\mathbf{x}_\tau$ and $\mathbf{p}_\tau$ are permuted tokens for $\mathbf{x}$ and $\mathbf{p}_{ta}$ w.r.t. to the permutation τ, respectively. By associating the target token's positional embedding with the next-token prediction, each token prediction is aware of the target token's index, alleviating the potential confusion in permuted objective.

Notably, we omit the target-aware positional embedding for the final token x_{τ_T}, as it does not participate in the loss computation and has no prediction target. A visual illustration of this concept is provided in Fig. 3. It is also noteworthy that the target-aware positional embedding only has impacts during training with permutation, and it can be merged with original positional embedding after the training is finished,

because our method anneals to a fixed raster scan in the end, and thus leads to no increase on the parameters or computation during inference.

Randomness Annealing. While the proposed randomized autoregressive training with permutation enables the model to capture bidirectional context within a unidirectional framework, it may introduce sub-optimal behavior for visual generation due to two main factors: (1) The sheer number of possible permutations is vast, potentially causing the model to focus on learning how to handle the different permutation orders rather than improving generation quality. For example, for a token sequence of length 256, the number of possible permutations is $256! > 10^{506}$, which can overwhelm the model and reduce training efficiency. (2) Although images can be processed in arbitrary orders, certain scan orders tend to outperform others. For instance, [22] evaluated six different scan orders (row-major, spiral in, spiral out, z-curve, subsample, and alternate) and found that row-major (*i.e.*, raster order) consistently performed the best, a result that has made it the most widely used order for visual generation.

To address these issues, we propose Randomness Annealing, a strategy designed to balance the randomness of permutations with the known effectiveness of the raster order. This method introduces a single parameter, r, which controls the probability of using a random permutation versus the raster order. At the start of training, $r = 1$, meaning that the model exclusively uses random permutations. Over the course of training, r linearly decays to 0, transitioning the model to the raster order by the end of training. Specifically, we define a training schedule for r, controlled by two hyper-parameters $start$ and end indicating the training epoch when r starts to anneal and when the annealing ends. Formally, we have:

$$r = \begin{cases} 1.0, & \text{if } epoch < start, \\ 0.0, & \text{if } epoch > end, \\ 1.0 - \frac{epoch - start}{end - start}, & \text{otherwise}, \end{cases} \tag{5}$$

where $epoch$ is the current training epoch. We will ablate the hyper-parameters $start$ and end in the experiments.

The schedule allows the model to initially explore the diverse random permutations for better bidirectional representation learning, and ultimately converge to the more effective row-major scan order for better visual generation quality, as is used by other typical autoregressive methods [22]. It is worth noting that this strategy not only improves generation performance but also maintains compatibility with the optimization techniques for standard scan orders used in previous works.

4. Experimental Results

In this section, we outline the implementation details of our method in Sec. 4.1. Next, we present ablation studies on key design choices in Sec. 4.2. The main results are discussed in Sec. 4.3, followed by scaling study and visualizations.

model	depth	width	mlp	heads	#params
RAR-B	24	768	3072	16	261M
RAR-L	24	1024	4096	16	461M
RAR-XL	32	1280	5120	16	955M
RAR-XXL	40	1408	6144	16	1499M

Table 1. **Architecture configurations of RAR.** We follow prior works scaling up ViT [19, 80] for different configurations.

4.1. Implementation Details

We implement the RAR on top of language modeling autoregressive framework with minimal changes.

VQ Tokenizer. Following prior works [10, 22] which use a VQ tokenizer to tokenize the input images into discrete token sequences, we use the MaskGIT-VQGAN [10] with the official weight trained on ImageNet. This tokenizer is a purely CNN-based tokenizer which tokenizes a 256×256 image into 256 discrete tokens (*i.e.*, downsampling factor 16) with a codebook size (*i.e.*, vocabulary size) 1024.

Autoregressive Transformer. We use vision transformers [19] of different model configurations [80] including RAR-S (133M), RAR-B (261M), RAR-L (461M), RAR-XL (955M), and RAR-XXL (1499M). For all of these model variants, we apply causal attention masking in the self-attention module and QK LayerNorm [15] to stabilize the large-scale model training. We use plain ViT for all ablation studies to speed up the experiments, and we enhance the model with adaLN [47] for final models. The detailed architecture configuration and model size are available at Tab. 1.

Positional Embedding. We use learnable embeddings for both original positional embedding in ViT and target-aware positional embedding. Notably, as our model anneals to raster order-based autoregressive image generation after the training is finished, the two positional embeddings can be combined into one, making it identical to a conventional autoregressive image generator.

Dataset. We train our model on ImageNet-1K [16] training set, which contains $1,281,167$ training images across 1000 object classes. We pre-tokenize the whole training set with MaskGIT-VQGAN tokenizer [10] to speed up the training. For ablation studies, we pre-tokenize the dataset with only center crop and horizontal flipping augmentation, while we further enhance the diversity in pretokenized datasets with ten-crop transformation [57, 58] for final models.

Training Protocols. We use the same training hyper-parameters for all model variants. The model is trained with batch size 2048 for 400 epochs ($250k$ steps). The learning rate will be linearly increased from 0 to 4×10^{-4} at the first 100 epochs (warm-up), then it will be gradually decayed to 1×10^{-5} following a cosine decay schedule.

start epoch	end epoch	FID↓	IS↑	Pre.↑	Rec.↑
0	0†	3.08	245.3	0.85	0.52
0	100	2.68	237.3	0.84	0.54
0	200	2.41	251.5	0.84	0.54
0	300	2.40	258.4	0.84	0.54
0	400	2.43	265.3	0.84	0.53
100	100	2.48	247.5	0.84	0.54
100	200	2.28	253.1	0.83	0.55
100	300	2.33	258.4	0.83	0.54
100	400	2.39	266.5	0.84	0.54
200	200	2.39	259.7	0.84	0.54
200	300	2.18	269.7	0.83	0.55
200	400	2.55	241.6	0.84	0.54
300	300	2.41	269.1	0.84	0.53
300	400	2.74	236.4	0.83	0.54
400	400‡	3.01	305.6	0.84	0.52

Table 2. **Different *start* and *end* epochs for randomness annealing, with a total of 400 training epochs and model size RAR-L.** The final setting is labeled in gray. †: When *start* epoch and *end* epoch are both 0 (1st row), the training reverts to a standard raster order training. ‡: When *start* epoch and *end* epoch are both 400 (last row), the training becomes a purely random order training. After training is finished, all results are obtained with raster order sampling, except for the purely random order training (*i.e.*, last row), where we also randomly sample the scan order following [38], which otherwise could not produce a reasonable result.

We use AdamW [35, 40] optimizer with beta1 0.9, beta2 0.96, and weight decay 0.03. We perform gradient clipping with maximum gradient norm 1.0. During training, the class condition will be dropped at a probability 0.1. The training setting remain the same for both ablation studies and main results across all RAR model variants.

Sampling Protocols. We sample 50000 images for FID computation using the evaluation code from [18]. We do not use any top-k or top-p based filtering techniques. We also follow prior arts [11, 25, 79] to use classifier-free guidance [31]. In the ablation study, we use a simpler linear guidance schedule [11], and for final models we use the improved power-cosine guidance schedule [25]. The final detailed hyper-parameters for each model variant can be found in the appendix.

4.2. Ablation Studies

We study different configurations for RAR, including the randomness annealing strategy and scan orders that RAR converges to.

Randomness Annealing Strategy. In Tab. 2 we compare different randomness annealing strategies. We adopt a linear decaying schedule and focus on when should the randomization annealing *starts* and *ends* by changing two hyperparameters *start* and *end*, as defined in Eq. (5). For a training lasting for 400 epochs, we enumerate all possible combinations for every 100 epochs. For example, when $start = 200$ and $end = 300$, the model is trained with random permutations from 0 to 200 epochs and raster order from 300 to 400 epochs. During 200 to 300 epoch, the model is trained via random permutation with probability r and raster order with probability $1 - r$, where r is computed as in Eq. (5). It is noteworthy that when $start = end = 0$, the model is trained with purely raster order, *i.e.*, the standard autoregressive training. When $start = end = 400$, the model is always trained with randomly permuted input sequence. Both cases are important baselines of the proposed randomness annealing, and they achieve FID scores of 3.08 and 3.01, respectively. Notably, pure random order training does not bring notable performance improvements. Interestingly, we observe all other variants with randomness annealing achieve substantial improvement over these two baselines. For example, even simply replacing the first 100 epochs of raster order with random permutation, it (*i.e.*, $start = 100$ and $end = 100$) improves the FID to 2.48 by 0.6. Besides, we also note that the model prefers to keep some beginning epochs for pure random permutation training and some last epochs for better adapting to raster scan order, which usually leads to a better performance compared to other variants. All the results demonstrate that adding randomized autoregressive training with a permuted objective is beneficial to the autoregressive visual generator and leads to a boosted FID score, thanks to the improved bidirectional representation learning process.

Additionally, among all variants, we found that the case, where $start = 200$ and $end = 300$, works the best, which improves the baseline (purely raster order) FID from 3.08 to 2.18. This strategy allocates slightly more computes on the training with random permutation order, and focuses on the purely raster order for the last 100 epochs. Therefore, we default to adopt this annealing strategy for all RAR models.

scan order	FID↓	IS↑	Precision↑	Recall↑
row-major	2.18	269.7	0.83	0.55
spiral in	2.50	256.1	0.84	0.54
spiral out	2.46	256.6	0.84	0.54
z-curve	2.29	262.7	0.83	0.55
subsample	2.39	258.0	0.84	0.54
alternate	2.48	270.9	0.84	0.53

Table 3. **Effect of different scan orders RAR-L converges to.** We mainly consider 6 different scan orders (row major, spiral in, spiral out, z-curve, subsample, alternate) as studied in [22]. Our default setting is marked in gray. A visual illustration of different scan orders are available in the appendix.

Different Scan Orders Besides Raster. Although row-major order (*i.e.*, raster scan) has been the de facto scan order in the visual generation, there lacks a systematic study on how good it is compared to other scan orders. We note that the work [22] conducted a similar study 4 years ago. However, it is worth re-examining the conclusion considering the significant progress generative models have achieved

in recent years. Specifically, we consider 6 different scan orders (row-major, spiral in, spiral out, z-curve, subsample, and alternative) following [22] that RAR may converge to. Instead of reporting the training loss and validation loss as the comparison metric [22], we directly evaluate their generation performance. The results are summarized in Tab. 3. Interestingly, we observe that all variants achieve a reasonably good score, which indicates that RAR is capable of handling different scan orders. Considering that the row-major (raster scan) still demonstrates advantages over the other scan orders, we thus use the raster scan order for all final RAR models.

We provide more ablation experiments regarding training epochs, different tokenizers *etc*. are available in the supplementary material.

4.3. Main Results

We report RAR results against state-of-the-art image generators on ImageNet-1K 256×256 benchmark [16].

As shown in Tab. 4, RAR achieves significantly better performance compared to previous AR image generators. Specifically, the most compact RAR-B with 261M parameters only, achieves an FID score 1.95, already significantly outperforming current state-of-the-art AR image generators LlamaGen-3B-384 (3.1B, FID 2.18, crop size 384) [57] and Open-MAGVIT2-XL (1.5B, FID 2.33) [41], while using 91% and 81% fewer model parameters respectively. It also surpasses the widely used diffusion models such as DiT-XL/2 (FID 1.95 *vs*. 2.27) and SiT-XL (FID 1.95 *vs*. 2.06) while only using 39% model parameters compared to them.

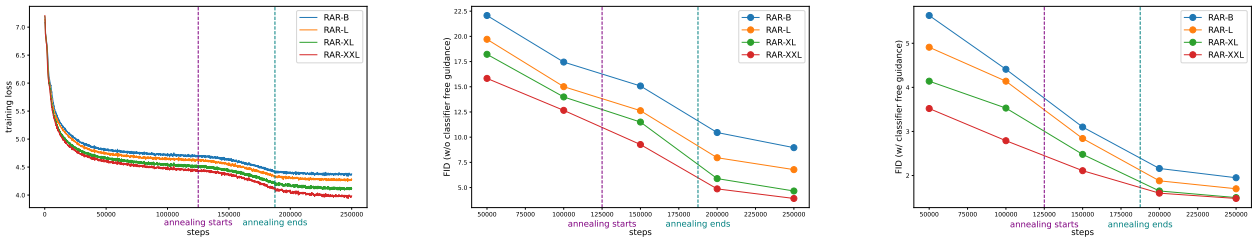
In Tab. 4, we further explore RAR at different model sizes (from 261M to 1.5B), where we observe strong scalability behavior with consistent performance improvement as model size scales up. Notably, the largest variant RAR-XXL sets a new state-of-the-art result on ImageNet benchmark, with an FID score 1.48. When compared to the other two recent methods VAR [63] and MAR [38], both of which attempt to amend AR formulation for better visual generation quality, RAR not only demonstrates a superior performance (FID 1.48 from RAR *vs*. 1.73 from VAR and 1.55 from MAR), but also keeps the whole framework compatible with language modeling and thus is more friendly for adapting the mature optimization and speed-up techniques for large language models to visual generation [57].

Moreover, RAR demonstrates superior performance to state-of-the-art visual generators in different frameworks. It performs better against the leading autoregressive models, diffusion models and masked transformer models, surpassing LlamaGen-3B-384 [57], MDTv2-XL/2 [25] and MaskBit [70] respectively (FID 1.48 from RAR *vs*. 2.18 from LlamaGen, 1.58 from MDTv2, and 1.52 from MaskBit). To the best of our knowledge, this is the first time that the language modeling style autoregressive visual generators outperform state-of-the-art diffusion models and masked transformer models.

tokenizer	type	generator	#params	FID↓	IS↑	Pre.↑	Rec.↑
VQ [54]	Diff.	LDM-8 [54]	258M	7.76	209.5	0.84	0.35
VAE [54]	Diff.	LDM-4 [54]	400M	3.60	247.7	0.87	0.48
VAE [56]	Diff.	UViT-L/2 [6]	287M	3.40	219.9	0.83	0.52
		UViT-H/2 [6]	501M	2.29	263.9	0.82	0.57
		DiT-L/2 [47]	458M	5.02	167.2	0.75	0.57
		DiT-XL/2 [47]	675M	2.27	278.2	0.83	0.57
		SiT-XL [42]	675M	2.06	270.3	0.82	0.59
		DiMR-XL/2R [39]	505M	1.70	289.0	0.79	0.63
		MDTv2-XL/2 [25]	676M	1.58	314.7	0.79	0.65
VQ [10]	Mask.	MaskGIT [10]	177M	6.18	182.1	-	-
VQ [79]	Mask.	TiTok-S-128 [79]	287M	1.97	281.8	-	-
VQ [78]	Mask.	MAGVIT-v2 [78]	307M	1.78	319.4	-	-
VQ [70]	Mask.	MaskBit [70]	305M	1.52	328.6	-	-
VAE [38]	MAR	MAR-B [38]	208M	2.31	281.7	0.82	0.57
		MAR-L [38]	479M	1.78	296.0	0.81	0.60
		MAR-H [38]	943M	1.55	303.7	0.81	0.62
VQ [63]	VAR	VAR-d30 [63]	2.0B	1.92	323.1	0.82	0.59
		VAR-d30-re [63]	2.0B	1.73	350.2	0.82	0.60
VQ [22]	AR	GPT2 [22]	1.4B	15.78	74.3	-	-
		GPT2-re [22]	1.4B	5.20	280.3	-	-
VQ [75]	AR	VIM-L [75]	1.7B	4.17	175.1	-	-
		VIM-L-re [75]	1.7B	3.04	227.4	-	-
VQ [41]	AR	Open-MAGVIT2-B [41]	343M	3.08	258.3	0.85	0.51
		Open-MAGVIT2-L [41]	804M	2.51	271.7	0.84	0.54
		Open-MAGVIT2-XL [41]	1.5B	2.33	271.8	0.84	0.54
VQ [57]	AR	LlamaGen-L [57]	343M	3.80	248.3	0.83	0.51
		LlamaGen-XL [57]	775M	3.39	227.1	0.81	0.54
		LlamaGen-XXL [57]	1.4B	3.09	253.6	0.83	0.53
		LlamaGen-3B [57]	3.1B	3.05	222.3	0.80	0.58
		LlamaGen-L-384 [57]	343M	3.07	256.1	0.83	0.52
		LlamaGen-XL-384 [57]	775M	2.62	244.1	0.80	0.57
		LlamaGen-XXL-384 [57]	1.4B	2.34	253.9	0.80	0.59
		LlamaGen-3B-384 [57]	3.1B	2.18	263.3	0.81	0.58
VQ [10]	AR	RAR-B (ours)	261M	1.95	290.5	0.82	0.58
		RAR-L (ours)	461M	1.70	299.5	0.81	0.60
		RAR-XL (ours)	955M	1.50	306.9	0.80	0.62
		RAR-XXL (ours)	1.5B	**1.48**	326.0	0.80	0.63

Table 4. **ImageNet-1K** 256×256 **generation results evaluated with ADM [18].** "type" refers to the type of the generative model, where "Diff." and "Mask." stand for diffusion models and masked transformer models, respectively. "VQ" denotes discrete tokenizers and "VAE" stands for continuous tokenizers. "-re" stands for rejection sampling. "-384" denotes for generating images at resolution 384 and resize back to 256 for evaluation, as is used in [57].

Sampling Speed. One key advantage of AR methods is their ability to leverage established optimization techniques from LLMs, such as KV-caching. In Tab. 5, we compare the sampling speed (measured as images/sec) of RAR against other types of generative models, such diffusion models [47], masked transformers [70, 79], VAR [63], and MAR [38]. Among them, AR models (RAR) and VAR models (VAR-d30) are compatible with the KV-cache optimization, providing a significant advantage in generation speed over other methods. As shown in Tab. 5, RAR achieves a state-of-



(a) training losses (b) FID scores w/o classifier-free guidance (c) FID scores w/ classifier-free guidance

Figure 4. **Scaling behavior of RAR models.** The scaled-up RAR models demonstrate (a) reduced training losses, and improved FID scores both (b) without and (c) with classifier-free guidance.

method	type	#params	FID↓	steps	images/sec
DiT-XL/2 [47]	Diff.	675M	2.27	250	0.6
TiTok-S-128 [79]	Mask.	287M	1.97	64	7.8
VAR-d30 [63]	VAR	2.0B	1.92	10	17.3
MAR-B [38]	MAR	208M	2.31	256	0.8
RAR-B (ours)	AR	261M	1.95	256	17.0
MAR-L [38]	MAR	479M	1.78	256	0.5
RAR-L (ours)	AR	461M	1.70	256	15.0
MaskBit [70]	Mask.	305M	1.52	256	0.7
MAR-H [38]	MAR	943M	1.55	256	0.3
RAR-XL (ours)	AR	955M	1.50	256	8.3
RAR-XXL (ours)	AR	1.5B	1.48	256	6.4

Table 5. **Sampling throughput comparison (including detokenization process) categorized by methods with similar FID scores.** Throughputs are measured as samples generated per second on a single A100 using float32 precision and a batch size of 128, based on their official codebases. For VAR [63] and our RAR, KV-cache is applied. "Diff." and "Mask." refer to diffusion models and masked transformer models, respectively.

Figure 5. **Visualization of samples generated by RAR across various model sizes.** RAR generates high-quality visual samples across all model sizes. As model size increases, fidelity and diversity improve, especially in challenging classes (*e.g.*, dogsled).

the-art FID score while also significantly surpassing other methods in generation speed. For instance, at an FID score around 1.5, MaskBit [70] and MAR-H [38] generate image samples at 0.7 and 0.3 images per second, respectively. In comparison, RAR-XL not only achieves a better FID score but can generate 8.3 high-quality visual samples per second—11.9× faster than MaskBit and 27.7× faster than MAR-H. The largest RAR variant, RAR-XXL, further improves the FID score while maintaining a notable speed advantage, being 9.1× faster than MaskBit and 21.3× faster than MAR-H. Additionally, RAR may benefit further from LLM optimization techniques such as vLLM [36], as seen with other AR methods [57].

Scaling Behavior. We study the scaling behavior of RAR. Specifically, we plot the training loss curves and FID score curves (with and without classifier-free guidance [31]) in Fig. 4. As shown in the figure, we observe that RAR scales well at different model sizes, where larger model size leads to a consistently lower training loss and better FID score, regardless of using the enhancement of classifier-free guidance or not. We note that as RAR keeps the AR formulation and framework intact, it also inherits the scalability from AR methods.

Visualization. We visualize generated samples by different RAR variants in Fig. 5, which shows that RAR is capable of generating high-quality samples with great fidelity and diversity. More visualizations are provided in the appendix.

5. Conclusion

In this paper, we introduced a simple yet effective strategy to enhance the visual generation quality of language modeling-compatible autoregressive image generators. By employing a randomized permutation objective, our approach enables improved bidirectional context learning while preserving the autoregressive structure. Consequently, the proposed RAR model not only surpasses previous state-of-the-art autoregressive image generation methods but also outperforms leading non-autoregressive transformer and diffusion models. We hope this research contributes to advancing autoregressive transformers toward a more powerful unified framework for visual understanding and generation.

References

[1] Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024. 1, 2

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

[3] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[4] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 2

[5] Yutong Bai, Xinyang Geng, Karttikeya Mangalam, Amir Bar, Alan Yuille, Trevor Darrell, Jitendra Malik, and Alexei A Efros. Sequential modeling enables scalable learning for large vision models. In *CVPR*, 2024. 1

[6] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *CVPR*, 2023. 1, 7

[7] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024. 1, 3

[8] Andrew Brock. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 3

[9] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 2020. 1, 2, 3

[10] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *CVPR*, 2022. 1, 2, 3, 4, 5, 7

[11] 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. In *ICML*, 2023. 3, 6

[12] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020. 2

[13] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023. 2

[14] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *JMLR*, 25(70):1–53, 2024. 1

[15] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *ICML*, pages 7480–7512. PMLR, 2023. 5

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2, 5, 7

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2018. 3

[18] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 2021. 1, 3, 6, 7

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 4, 5

[20] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 2

[21] Alaaeldin El-Nouby, Michal Klein, Shuangfei Zhai, Miguel Angel Bautista, Alexander Toshev, Vaishaal Shankar, Joshua M Susskind, and Armand Joulin. Scalable pre-training of large autoregressive image models. *ICML*, 2024. 1, 3

[22] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 1, 2, 3, 4, 5, 6, 7

[23] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024. 3

[24] Lijie Fan, Tianhong Li, Siyang Qin, Yuanzhen Li, Chen Sun, Michael Rubinstein, Deqing Sun, Kaiming He, and Yonglong Tian. Fluid: Scaling autoregressive text-to-image generative models with continuous tokens. *arXiv preprint arXiv:2410.13863*, 2024. 3

[25] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer. In *ICCV*, 2023. 6, 7

[26] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NeurIPS*, 2014. 3

[27] Karol Gregor, Ivo Danihelka, Andriy Mnih, Charles Blundell, and Daan Wierstra. Deep autoregressive networks. In *ICML*, 2014. 2

[28] Jiatao Gu, Changhan Wang, and Junbo Zhao. Levenshtein transformer. *NeurIPS*, 32, 2019. 4

[29] Ju He, Qihang Yu, Qihao Liu, and Liang-Chieh Chen. Flowtok: Flowing seamlessly across text and image tokens. In *ICCV*, 2025. 1

[30] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 1

[31] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 6, 8

[32] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. 3

[33] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 3

[34] Dongwon Kim, Ju He, Qihang Yu, Chenglin Yang, Xiaohui Shen, Suha Kwak, and Liang-Chieh Chen. Democratizing text-to-image masked generative models with compact text-aware one-dimensional tokens. In *ICCV*, 2025. 1

[35] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[36] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626, 2023. 8

[37] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *CVPR*, 2022. 1

[38] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *NeurIPS*, 2024. 1, 2, 3, 6, 7, 8

[39] Qihao Liu, Zhanpeng Zeng, Ju He, Qihang Yu, Xiaohui Shen, and Liang-Chieh Chen. Alleviating distortion in image generation via multi-resolution diffusion models. *NeurIPS*, 2024. 3, 7

[40] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *ICLR*, 2019. 6

[41] 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. 1, 2, 7

[42] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *ECCV*, 2024. 7

[43] Xiaoxiao Ma, Mohan Zhou, Tao Liang, Yalong Bai, Tiejun Zhao, Huaian Chen, and Yi Jin. Star: Scale-wise text-to-image generation via auto-regressive representations. *arXiv preprint arXiv:2406.10797*, 2024. 3

[44] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *NeurIPS*, 36, 2023. 4

[45] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1, 2

[46] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *ICML*, 2018. 2

[47] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 1, 3, 5, 7, 8

[48] Alec Radford. Improving language understanding by generative pre-training. *OpenAI*, 2018. 2

[49] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2

[50] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021. 2

[51] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *NeurIPS*, 2019. 2

[52] Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Beyond next-token: Next-x prediction for autoregressive visual generation. In *ICCV*, 2025. 1

[53] Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Flowar: Scale-wise autoregressive image generation meets flow matching. In *ICML*, 2025. 1

[54] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 3, 7

[55] Inkyu Shin, Chenglin Yang, and Liang-Chieh Chen. Deeply supervised flow-based generative models. In *ICCV*, 2025. 1

[56] stabilityai, 2023. 7

[57] 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. 1, 2, 4, 5, 7, 8

[58] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 5

[59] 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. 3

[60] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024. 1, 2

[61] Emu3 Team. Emu3: Next-token prediction is all you need. *Tech Report*, 2024. 1, 2

[62] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 1, 2

[63] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *NeurIPS*, 2024. 1, 2, 3, 7, 8

[64] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama:

Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1, 2

[65] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1, 2

[66] Benigno Uria, Marc-Alexandre Côté, Karol Gregor, Iain Murray, and Hugo Larochelle. Neural autoregressive distribution estimation. *JMLR*, 17(205):1–37, 2016. 3

[67] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *NeurIPS*, 2016. 2

[68] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *ICML*, 2016. 2

[69] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *NeurIPS*, 2017. 2

[70] Mark Weber, Lijun Yu, Qihang Yu, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. Maskbit: Embedding-free image generation via bit tokens. *arXiv preprint arXiv:2409.16211*, 2024. 1, 3, 7, 8

[71] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *ICLR*, 2022. 1

[72] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. 1, 2

[73] Chenglin Yang, Celong Liu, Xueqing Deng, Dongwon Kim, Xing Mei, Xiaohui Shen, and Liang-Chieh Chen. 1.58-bit flux. *arXiv preprint arXiv:2412.18653*, 2024. 3

[74] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *NeurIPS*, 2019. 3

[75] 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. In *ICLR*, 2022. 1, 2, 7

[76] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *TMLR*, 2022. 1

[77] 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 *CVPR*, 2023. 1, 3

[78] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, et al. Language model beats diffusion–tokenizer is key to visual generation. In *ICLR*, 2024. 3, 7

[79] Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. *NeurIPS*, 2024. 1, 3, 6, 7, 8

[80] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *CVPR*, pages 12104–12113, 2022. 5

[81] 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. 3