Visual Autoregressive Modeling: Scalable Image Generation via Next-Scale Prediction

Keyu Tian[1,2], **Yi Jiang**[2,†], **Zehuan Yuan**[2,*], **Bingyue Peng**[2], **Liwei Wang**[1,*]
[1]Peking University [2]Bytedance Inc
keyutian@stu.pku.edu.cn, jiangyi.enjoy@bytedance.com,
yuanzehuan@bytedance.com, bingyue.peng@bytedance.com, wanglw@pku.edu.cn

Try and explore our online demo at: https://var.vision
Codes and models: https://github.com/FoundationVision/VAR

Figure 1: **Generated samples from Visual AutoRegressive (VAR) transformers trained on ImageNet**. We show 512×512 samples (top), 256×256 samples (middle), and zero-shot image editing results (bottom).

Abstract

We present Visual AutoRegressive modeling (VAR), a new generation paradigm that redefines the autoregressive learning on images as coarse-to-fine "next-scale prediction" or "next-resolution prediction", diverging from the standard raster-scan "next-token prediction". This simple, intuitive methodology allows autoregressive (AR) transformers to learn visual distributions fast and can generalize well: VAR, for the *first time*, makes GPT-style AR models surpass diffusion transformers in image generation. On ImageNet 256×256 benchmark, VAR significantly improve AR baseline by improving Fréchet inception distance (FID) from 18.65 to 1.80, inception score (IS) from 80.4 to 356.4, with 20× faster inference speed. It is also empirically verified that VAR outperforms the Diffusion Transformer (DiT) in multiple dimensions including image quality, inference speed, data efficiency, and scalability. Scaling up VAR models exhibits clear power-law scaling laws similar to those observed in LLMs, with linear correlation coefficients near −0.998 as solid evidence. VAR further showcases zero-shot generalization ability in downstream tasks including image in-painting, out-painting, and editing. These results suggest VAR has initially emulated the two important properties of LLMs: **Scaling Laws** and **zero-shot** generalization. We have released all models and codes to promote the exploration of AR/VAR models for visual generation and unified learning.

*Corresponding authors: wanglw@pku.edu.cn, yuanzehuan@bytedance.com; †: project lead

Preprint version.

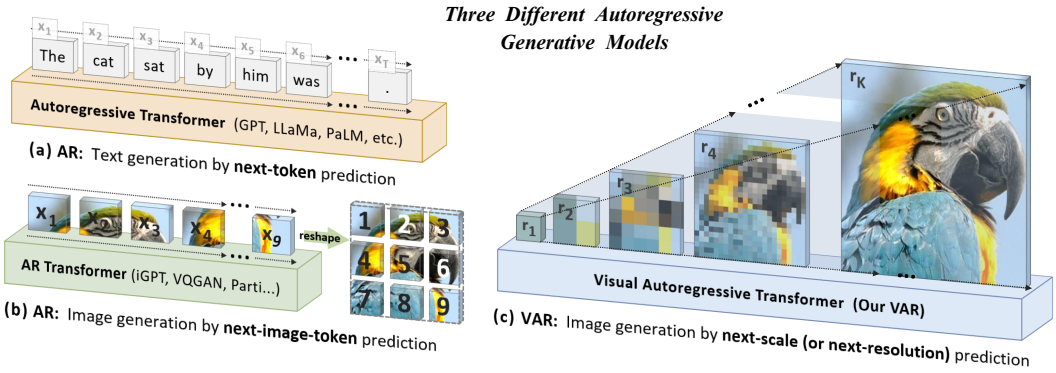


Figure 2: **Standard autoregressive modeling (AR) *vs.* our proposed visual autoregressive modeling (VAR).** (a) AR applied to language: sequential text token generation from left to right, word by word; (b) AR applied to images: sequential visual token generation in a raster-scan order, from left to right, top to bottom; (c) VAR for images: multi-scale token maps are autoregressively generated from coarse to fine scales (lower to higher resolutions), with parallel token generation within each scale. VAR requires a multi-scale VQVAE to work.

1 Introduction

The advent of GPT series [48, 49, 9, 45, 1] and other autoregressive (AR) large language models (LLMs) [13, 3, 27, 63, 64, 70, 60, 4, 61] has heralded a new epoch in the field of artificial intelligence. These models exhibit promising intelligence in generality and versatility that, despite issues like hallucinations [28], are still considered to take a solid step toward the general artificial intelligence (AGI). The crux behind these large models is a self-supervised learning strategy – *predicting the next token* in a sequence, a simple yet profound approach. Studies into **the success of these large AR models** have highlighted their **scalability and generalizabilty**: the former, as exemplified by *scaling laws* [30, 22], allows us to predict large model's performance from smaller ones and thus guides better resource allocation, while the latter, as evidenced by *zero-shot and few-shot* learning [49, 9], underscores the unsupervised-trained models' adaptability to diverse, unseen tasks. These properties reveal AR models' potential in learning from vast unlabeled data, encapsulating the essence of "AGI".

In parallel, the field of computer vision has been striving to develop large autoregressive or world models [42, 68, 5], aiming to emulate their impressive scalability and generalizability. Trailblazing efforts like VQGAN and DALL-E [19, 51] along with their successors [52, 71, 37, 77] have showcased the potential of AR models in image generation. These models utilize a visual tokenizer to discretize continuous images into grids of 2D tokens, which are then flattened to a 1D sequence for AR learning (Fig. 2 b), mirroring the process of sequential language modeling (Fig. 2 a). However, the scaling laws of these models remain underexplored, and more frustratingly, their performance **significantly lags** behind diffusion models [46, 2, 38], as shown in Fig. 3. In contrast to the remarkable achievements of LLMs, the power of autoregressive models in computer vision appears to be somewhat **locked**.

Autoregressive modeling requires defining the order of data. Our work reconsiders how to "order" an image. Humans typically perceive or create images in a hierachical manner, first capturing the global structure and then local details. This multi-scale, coarse-to-fine method naturally suggests an "order" for images. Also inspired by the widespread multi-scale designs [40, 39, 62, 31, 33], we define autoregressive learning for images as "next-scale prediction" in Fig. 2 (c), diverging from the conventional "next-token prediction" in Fig. 2 (b). Our approach begins by encoding an image into multi-scale token maps. The autoregressive process is then started from the 1×1 token map, and progressively expands in resolution: at each step, the transformer predicts the next higher-resolution token map conditioned on all previous ones. We refer to this methodology as Visual AutoRegressive (VAR) modeling.

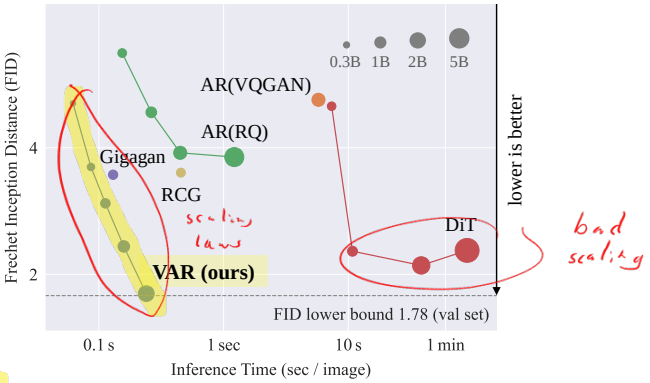


Figure 3: **Scaling behavior** of different models on ImageNet 256×256 conditional generation benchmark. The FID of the validation set serves as a reference lower bound (1.78). VAR with 2B parameters reaches FID 1.80, surpassing L-DiT with 3B or 7B parameters.

VAR directly leverages GPT-2-like transformer architecture [49] for visual autoregressive learning. On the ImageNet 256×256 benchmark, VAR significantly improves its AR baseline, achieving a Fréchet inception distance (FID) of 1.80 and an inception score (IS) of 356.4, with inference speed 20× faster (see Sec. 4.4 for details). Notably, VAR surpasses the Diffusion Transformer (DiT) – the foundation of leading diffusion systems like Stable Diffusion 3.0 and SORA [18, 8] – in FID/IS, data efficiency, inference speed, and scalability. VAR models also exhibit scaling laws akin to those witnessed in LLMs. Lastly, we showcase VAR's zero-shot generalization capabilities in tasks like image in-painting, out-painting, and editing. In summary, our contributions to the community include:

1. A new visual generative framework using a multi-scale autoregressive paradigm with next-scale prediction, offering new insights in autoregressive algorithm design for computer vision.
2. An empirical validation of VAR models' Scaling Laws and zero-shot generalization potential, which initially emulates the appealing properties of large language models (LLMs).
3. A breakthrough in visual autoregressive model performance, making GPT-style autoregressive methods surpass strong diffusion models in image synthesis *for the first time*[2].
4. A comprehensive open-source code suite, including both VQ tokenizer and autoregressive model training pipelines, to help propel the advancement of visual autoregressive learning.

2 Related Work

2.1 Properties of large autoregressive language models

Scaling laws. Power-law scaling laws [22, 30] mathematically describe the relationship between the growth of model parameters, dataset sizes, computational resources, and the performance improvements of machine learning models, conferring several distinct benefits. First, they facilitate the extrapolation of a larger model's performance through the scaling up of model size, data size, and computational cost. This helps save unnecessary costs and provides principles to allocate the training budget. Second, the scaling laws have evidenced a consistent and non-saturating increase in performance, corroborating their sustained advantage in enhancing model capability. Propelled by the principles of scaling laws in neural language models [30], several Large Language Models [9, 76, 70, 27, 63, 64] have been proposed, embodying the principle that increasing the scale of models tends to yield enhanced performance outcomes. GPT [49, 9], predicated on a transformer decoder architecture, undergoes generative pre-training and scales the model size to an unprecedented 175B parameters. LLama [63, 64] release a collection of pre-trained and fine-tuned large language models (LLMs) ranging in scale from 7 billion to 70 billion parameters. The manifest efficacy of scaling laws applied to language models has proffered a glimpse into the promising potential of upscale in visual models [5].

Zero-shot generalization. Zero-shot generalization [55] refers to the ability of a model, particularly a Large Language Model, to perform tasks that it has not been explicitly trained on. Within the realm of the vision community, there is a burgeoning interest in the zero-shot and in-context learning abilities of foundation models, CLIP [47], SAM [35], Dinov2 [44]. Innovations like Painter [69] and LVM [5] have leveraged visual prompts to design in-context learning paradigms, thereby facilitating the generalization to downstream unseen tasks.

2.2 Visual generation

Image tokenizer and autoregressive models. Language models [15, 49, 63] rely on Byte Pair Encoding (BPE [20]) or WordPiece algorithms for text tokenization. Visual generation models based on language models also necessitate the encoding of 2D images into 1D token sequences. Early endeavors VQVAE [65] have demonstrated the ability to represent images as discrete tokens, although the reconstruction quality was relatively moderate. VQGAN [19] advances VQVAE by incorporating adversarial loss and perceptual loss to improve image fidelity, and employs a decoder-only transformer to generate image tokens in standard raster-scan autoregressive manner. VQVAE-2 [52] and RQ-Transformer [37] also follow VQGAN's raster-scan autoregressive method but further improve VQVAE via extra scales or stacked codes. Parti [72] capitalizes on the foundational architecture of ViT-VQGAN [71] to scale the transformer model size to 20 billion parameters, achieving remarkable results in text-to-image synthesis.

VQVAE as an image tokenizer

[2] A related work [74] named "language model beats diffusion" belongs to BERT-style masked-prediction model.

Masked-prediction model. MaskGIT [11] employs a masked prediction framework [15, 6, 21] alongside a VQ autoencoder to generate image tokens based through a "greedy" algorithm. MagViT [73] adapts this approach to video data, and MagViT-2 [74] enhances MaskGIT by introducing an improved VQVAE. MUSE [10] scales MaskGIT's architecture to 3 billion parameters and merges it with the T5 language model [50], setting new benchmarks in text-to-image synthesis.

Diffusion models [23, 59, 16, 53] are considered the forefront of visual synthesis, given their superior generation quality and diversity. Progress in diffusion models has centered around improved sampling techniques[26], faster sampling [58, 41], and architectural enhancements [53, 24, 54, 46]. Imagen [54] incorporates the T5 language model [50] for text condition and builds an image generation system through multiple independent diffusion models for cascaded generation and super resolution. Latent Diffusion Models (LDM)[53] apply diffusion in latent space, improving efficiency in training and inference. DiT[46] replaces the traditional U-Net with a transformer-based architecture [66, 17], and is used in recent image or video synthesis systems like Stable Diffusion 3.0 [18] and SORA [8].

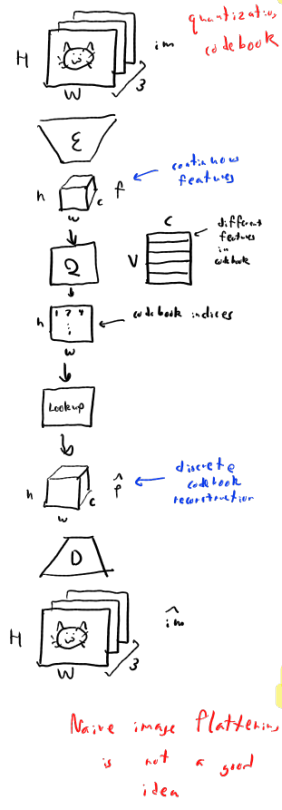
3 Method

3.1 Preliminary: autoregressive modeling via next-token prediction

Formulation. Consider a sequence of discrete tokens $x = (x_1, x_2, \ldots, x_T)$, where each token $x_t \in [V]$ is an integer from a vocabulary of size V. The next-token autoregressive model posits that the probability of observing the current token x_t depends only on its prefix $(x_1, x_2, \ldots, x_{t-1})$. This assumption of **unidirectional token dependency** allows us to decompose the likelihood of sequence x into the product of T conditional probabilities:

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

Training an autoregressive model p_θ parameterized by θ involves optimizing the $p_\theta(x_t \mid x_1, x_2, \ldots, x_{t-1})$ across a dataset. This optimization process is known as the "next-token prediction", and the trained p_θ can generate new sequences.



Tokenization. Images are inherently 2D continuous signals. To apply autoregressive modeling to images via next-token prediction, we must: 1) tokenize an image into several *discrete* tokens, and 2) define a 1D *order* of tokens for unidirectional modeling. For 1), a quantized autoencoder such as [19] is often used to convert the image feature map $f \in \mathbb{R}^{h \times w \times C}$ to discrete tokens $q \in [V]^{h \times w}$:

$$f = \mathcal{E}(im), \qquad q = \mathcal{Q}(f), \tag{2}$$

where im denotes the raw image, $\mathcal{E}(\cdot)$ a encoder, and $\mathcal{Q}(\cdot)$ a quantizer. The quantizer typically includes a learnable codebook $Z \in \mathbb{R}^{V \times C}$ containing V vectors. The quantization process $q = \mathcal{Q}(f)$ will map each feature vector $f^{(i,j)}$ to the code index $q^{(i,j)}$ of its nearest code in the Euclidean sense:

$$q^{(i,j)} = \left(\arg\min_{v \in [V]} \|\text{lookup}(Z, v) - f^{(i,j)}\|_2 \right) \in [V], \tag{3}$$

where $\text{lookup}(Z, v)$ means taking the v-th vector in codebook Z. To train the quantized autoencoder, Z is looked up by every $q^{(i,j)}$ to get $\hat{f}$, the approximation of original f. Then a new image $\hat{im}$ is reconstructed using the decoder $\mathcal{D}(\cdot)$ given $\hat{f}$, and a compound loss $\mathcal{L}$ is minimized:

$$\hat{f} = \text{lookup}(Z, q), \qquad \hat{im} = \mathcal{D}(\hat{f}), \tag{4}$$

$$\mathcal{L} = \|im - \hat{im}\|_2 + \|f - \hat{f}\|_2 + \lambda_P \mathcal{L}_P(\hat{im}) + \lambda_G \mathcal{L}_G(\hat{im}), \tag{5}$$

where $\mathcal{L}_P(\cdot)$ is a perceptual loss such as LPIPS [75], $\mathcal{L}_G(\cdot)$ a discriminative loss like StyleGAN's discriminator loss [33], and λ_P, λ_G are loss weights. Once the autoencoder $\{\mathcal{E}, \mathcal{Q}, \mathcal{D}\}$ is fully trained, it will be used to tokenize images for subsequent training of a unidirectional autoregressive model.

The image tokens in $q \in [V]^{h \times w}$ are arranged in a 2D grid. Unlike natural language sentences with an inherent left-to-right ordering, the order of image tokens must be explicitly defined for unidirectional autoregressive learning. Previous AR methods [19, 71, 37] flatten the 2D grid of q into a 1D sequence $x = (x_1, \ldots, x_{h \times w})$ using some strategy such as row-major raster scan, spiral, or z-curve order. Once flattened, they can extract a set of sequences x from the dataset, and then train an autoregressive model to maximize the likelihood in (1) via next-token prediction.

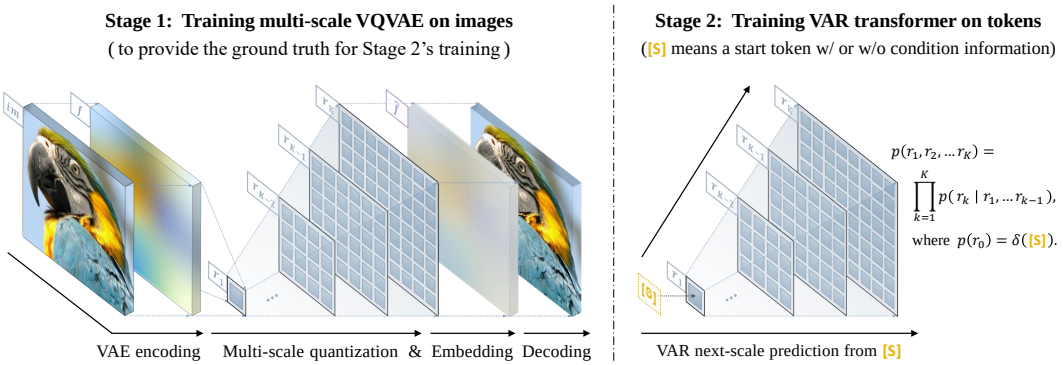


Figure 4: **VAR involves two separated training stages**. Left: a multi-scale quantized autoencoder (VQVAE) encode an image into K token maps $R = (r_1, r_2, \dots, r_K)$ and is trained by a compound reconstruction loss (5). "Embedding" in the figure means converting discrete tokens into continuous embedding vectors. Right: a VAR transformer is trained by maximizing the likelihood or minimizing the cross-entropy loss in (6) via next-scale prediction. the trained VQVAE is used to produce token map ground truth $R = (r_1, r_2, \dots, r_K)$ for VAR model.

Discussion. The above tokenizing and flattening enable next-token autoregressive learning on images, but they introduce several issues:

1) **Mathematical premise violation.** Image encoders typically produce image feature map f with inter-dependent feature vectors $f^{(i,j)}$ for all i, j. So after quantization and flattening, the sequence of tokens $(x_1, x_2, \dots, x_{h\times w})$ exhibits *bidirectional* correlations. This contradicts the *unidirectional* dependency assumption of autoregressive models, which dictates that each token x_t should only depend on its prefix $(x_1, x_2, \dots, x_{t-1})$.
2) **Structural degradation.** The flattening disrupts the spatial locality inherent in image feature maps. For instance, the token $q^{(i,j)}$ and its 4 immediate neighbors $q^{(i\pm1,j)}$, $q^{(i,j\pm1)}$ are closely correlated due to their proximity. This spatial relationship is compromised in the linear sequence x, where *uni*directional constraints diminish these *cor*relations.
3) **Inefficiency.** Generating an image token sequence $x = (x_1, x_2, \dots, x_{n\times n})$ with a conventional self-attention transformer incurs $\mathcal{O}(n^2)$ autoregressive steps and $\mathcal{O}(n^6)$ computational cost.

The disruption of spatial locality (issue 2) is obvious. Regarding issue 1, we present empirical evidence in the Appendix, analyzing the token dependencies in the popular quantized autoencoder [19] and revealing significant bidirectional correlations. The proof of computational complexity for issue 3 is detailed in the Appendix. These theoretical and practical limitations call for a rethinking of autoregressive models in the context of image generation.

3.2 Visual autoregressive modeling via next-scale prediction

Reformulation. We reconceptualize the autoregressive modeling on images by shifting from "next-token prediction" to "next-scale prediction" strategy. Here, the autoregressive unit is *an entire token map*, rather than *a single token*. We start by quantizing a feature map $f \in \mathbb{R}^{h\times w\times C}$ into K multi-scale token maps $(r_1, r_2, \dots, r_K)$, each at a increasingly higher resolution $h_k \times w_k$, culminating in r_K matches the original feature map's resolution $h \times w$. The autoregressive likelihood is formulated as:

$$p(r_1, r_2, \dots, r_K) = \prod_{k=1}^{K} p(r_k \mid r_1, r_2, \dots, r_{k-1}), \tag{6}$$

where each autoregressive unit $r_k \in [V]^{h_k\times w_k}$ is the token map at scale k, and the sequence $(r_1, r_2, \dots, r_{k-1})$ serves as the the "prefix" for r_k. During the k-th autoregressive step, all distributions over the $h_k \times w_k$ tokens in r_k are inter-dependent and will be generated in parallel, conditioned on r_k's prefix and associated k-th position embedding map. This "next-scale prediction" methodology is what we define as visual autoregressive modeling (VAR), depicted on the right side of Fig. 4.

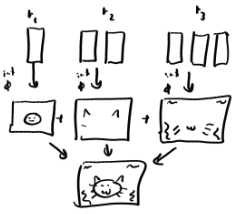
Discussion. VAR addresses the previously mentioned three issues as follows:

1) The mathematical premise is satisfied if we constrain each r_k to depend only on its prefix, that is, the process of getting r_k is solely related to $(r_1, r_2, \dots, r_{k-1})$. This constraint is acceptable as it aligns with the natural, coarse-to-fine progression characteristics like human visual perception and artistic drawing. Further details are provided in the *Tokenization* below.

2) The spatial locality is preserved as (i) there is no flattening operation in VAR, and (ii) tokens in each r_k are fully correlated. The multi-scale design additionally reinforces the spatial structure.

3) The complexity for generating an image with $n \times n$ latent is significantly reduced to $\mathcal{O}(n^4)$, see Appendix for proof. This efficiency gain arises from the *parallel* token generation in each r_k.

Tokenization. We develope a new multi-scale quantization autoencoder to encode an image to K multi-scale discrete token maps $R = (r_1, r_2, \ldots, r_K)$ necessary for VAR learning (6). We employ the same architecture as VQGAN [19] but with a modified multi-scale quantization layer. The encoding and decoding procedures with residual design on f or $\hat{f}$ are detailed in algorithms 1 and 2. We empirically find this residual-style design, akin to [37], can perform better than independent interpolation. Algorithm 1 shows that each r_k would depend only on its prefix $(r_1, r_2, \ldots, r_{k-1})$. Note that a shared codebook Z is utilized across all scales, ensuring that each r_k's tokens belong to the same vocabulary $[V]$. To address the information loss in upscaling z_k to $h_K \times w_K$, we use K extra convolution layers $\{\phi_k\}_{k=1}^{K}$. No convolution is used after downsampling f to $h_k \times w_k$.



Algorithm 1: Multi-scale VQVAE Encoding

Inputs: raw image im;
Hyperparameters: steps K, resolutions $(h_k, w_k)_{k=1}^{K}$;
$f = \mathcal{E}(im)$, $R = []$;
for $k = 1, \cdots, K$ **do**
  $r_k = \mathcal{Q}(\text{interpolate}(f, h_k, w_k))$;
  $R = \text{queue\_push}(R, r_k)$;
  $z_k = \text{lookup}(Z, r_k)$;
  $z_k = \text{interpolate}(z_k, h_K, w_K)$;
  $f = f - \phi_k(z_k)$;
Return: multi-scale tokens R;

Algorithm 2: Multi-scale VQVAE Reconstruction

Inputs: multi-scale token maps R;
Hyperparameters: steps K, resolutions $(h_k, w_k)_{k=1}^{K}$;
$\hat{f} = 0$;
for $k = 1, \cdots, K$ **do**
  $r_k = \text{queue\_pop}(R)$;
  $z_k = \text{lookup}(Z, r_k)$;
  $z_k = \text{interpolate}(z_k, h_K, w_K)$;
  $\hat{f} = \hat{f} + \phi_k(z_k)$;
$\hat{im} = \mathcal{D}(\hat{f})$;
Return: reconstructed image $\hat{im}$;

3.3 Implementation details

VAR tokenizer. As aforementioned, we use the vanilla VQVAE architecture [19] with a multi-scale quantization scheme with K extra convolutions (0.03M extra parameters). We use a shared codebook for all scales with $V = 4096$ and a latent dim of 32. Following the baseline [19], our tokenizer is also trained on OpenImages [36] with the compound loss (5). See the Appendix for more details.

VAR transformer. Our main focus is on VAR algorithm so we keep a simple model architecture design. We adopt the architecture of standard decoder-only transformers akin to GPT-2 and VQGAN [49, 19], with the only modification of substituting traditional layer normalization for adaptive normalization (AdaLN) – a choice motivated by its widespread adoption and proven effectiveness in visual generative models [33, 34, 32, 57, 56, 29, 46, 12]. For class-conditional synthesis, we use the class embedding as the start token `[s]` and also the condition of AdaLN. We do not use advanced techniques in modern large language models, such as rotary position embedding (RoPE), SwiGLU MLP, or RMS Norm [63, 64]. Our model shape hyperparameter follows a simple rule [30] that the width w, head counts h, and drop rate dr are linearly scaled with the depth d as follows:

$$w = 64d, \qquad h = d, \qquad dr = 0.1 \cdot d/24. \tag{7}$$

Consequently, the main parameter count N of a VAR transformer with depth d is given by[3]:

$$N(d) = \underbrace{d \cdot 4w^2}_{\text{self-attention}} + \underbrace{d \cdot 8w^2}_{\text{feed-forward}} + \underbrace{d \cdot 6w^2}_{\text{adaptive layernorm}} = 18\, dw^2 = 73728\, d^3. \tag{8}$$

All models are trained with the similar settings: a base learning rate of 10^{-4} per 256 batch size, an AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.95$, decay $= 0.05$, a batch size from 512 to 1024 and training epochs from 200 to 350 (depends on model size). Our subsequent evaluations in Sec. 4 will suggest that such a simple model design are capable of scaling and generalizing well.

[3]Due to resource limitation, we use a single shared adaptive layernorm (AdaLN) acorss all attention blocks in 512×512 synthesis. In this case, the parameter count would be reduced to around $12dw^2 + 6w^2 \approx 49152\, d^3$.

Table 1: **Generative model family comparison on class-conditional ImageNet 256×256**. "↓" or "↑" indicate lower or higher values are better. Metrics include Fréchet inception distance (FID), inception score (IS), precision (Pre) and recall (rec). "#Step": the number of model runs needed to generate an image. Wall-clock inference time relative to VAR is reported. Models with the suffix "-re" used rejection sampling. †: taken from MaskGIT [11].

Type	Model	FID↓	IS↑	Pre↑	Rec↑	#Para	#Step	Time
GAN	BigGAN [7]	6.95	224.5	**0.89**	0.38	112M	1	–
GAN	GigaGAN [29]	3.45	225.5	0.84	**0.61**	569M	1	–
GAN	StyleGan-XL [57]	2.30	265.1	0.78	0.53	166M	1	0.3 [57]
Diff.	ADM [16]	10.94	101.0	0.69	0.63	554M	250	168 [57]
Diff.	CDM [25]	4.88	158.7	–	–	–	8100	–
Diff.	LDM-4-G [53]	3.60	247.7	–	–	400M	250	–
Diff.	DiT-L/2 [46]	5.02	167.2	0.75	0.57	458M	250	31
Diff.	DiT-XL/2 [46]	2.27	278.2	0.83	0.57	675M	250	45
Diff.	L-DiT-3B [2]	2.10	304.4	0.82	0.60	3.0B	250	>45
Diff.	L-DiT-7B [2]	2.28	316.2	0.83	0.58	7.0B	250	>45
Mask.	MaskGIT [11]	6.18	182.1	0.80	0.51	227M	8	0.5 [11]
Mask.	MaskGIT-re [11]	4.02	355.6	–	–	227M	8	0.5 [11]
Mask.	RCG (cond.) [38]	3.49	215.5	–	–	502M	20	1.9 [38]
AR	VQVAE-2† [52]	31.11	∼45	0.36	0.57	13.5B	5120	–
AR	VQGAN† [19]	18.65	80.4	0.78	0.26	227M	256	19 [11]
AR	VQGAN [19]	15.78	74.3	–	–	1.4B	256	24
AR	VQGAN-re [19]	5.20	280.3	–	–	1.4B	256	24
AR	ViTVQ [71]	4.17	175.1	–	–	1.7B	1024	>24
AR	ViTVQ-re [71]	3.04	227.4	–	–	1.7B	1024	>24
AR	RQTran. [37]	7.55	134.0	–	–	3.8B	68	21
AR	RQTran.-re [37]	3.80	323.7	–	–	3.8B	68	21
VAR	VAR-$d16$	3.60	257.5	0.85	0.48	310M	10	0.4
VAR	VAR-$d20$	2.95	306.1	0.84	0.53	600M	10	0.5
VAR	VAR-$d24$	2.33	320.1	0.82	0.57	1.0B	10	0.6
VAR	VAR-$d30$	1.97	334.7	0.81	**0.61**	2.0B	10	1
VAR	VAR-$d30$-re	**1.80**	**356.4**	0.83	0.57	2.0B	10	1
	(validation data)	1.78	236.9	0.75	0.67			

4 Empirical Results

This section first compares VAR with other image generative model families[4] in terms of performance and and efficiency in Sec. 4.1. Evaluations on the scalability and generalizability of VAR models are presented in Sec. 4.2 and Sec. 4.3. We do some ablations and visualizations at the end.

4.1 State-of-the-art image generation

Setup. We test VAR models with depths 16, 20, 24, and 30 on ImageNet 256×256 and 512×512 conditional generation benchmarks and compare them with the state-of-the-art image generation model families. Among all VQVAE-based AR or VAR models, VQGAN [19] and ours use the same architecture (CNN) and training data (OpenImages [36]) for VQVAE, while ViT-VQGAN [71] uses a ViT autoencoder, and both it and RQTransformer [37] trains the VQVAE directly on ImageNet. The results are summaried in Tab. 1 and Tab. 2.

Overall comparison. In comparison with existing generative approaches including generative adversarial networks (GAN), diffusion models (Diff.), BERT-style masked-prediction models (Mask.), and GPT-style autoregressive models (AR), our visual autoregressive (VAR) establishes a new model class. As shown in Tab. 1, VAR not only achieves the best FID/IS but also demonstrates remarkable speed in image generation. VAR also maintains decent precision and recall, confirming its semantic consistency. These advantages hold true on the 512×512 synthesis benchmark, as detailed in Tab. 2. Notably, VAR significantly advances traditional AR capabilities. To our knowledge, this is the *first time* of autoregressive models outperforming Diffusion transformers, a milestone made possible by VAR's resolution of AR limitations discussed in Section 3.

[4]For fairness, methods with advanced VQVAE are excluded in this *model family comparison*, e.g. the BERT-style masked-pred model MagViT-2 [74]. Practitioners can combine them with VAR for better results.

Efficiency comparison. Conventional autoregressive (AR) models [19, 52, 71, 37] suffer a lot from the high computational cost, as the number of image tokens is quadratic to the image resolution. A full autoregressive generation of n^2 tokens requires $\mathcal{O}(n^2)$ decoding iterations and $\mathcal{O}(n^6)$ total computations. In contrast, VAR only requires $\mathcal{O}(\log(n))$ iterations and $\mathcal{O}(n^4)$ total computations. The wall-clock time reported in Tab. 1 also provides empirical evidence that VAR is around 20 times faster than VQGAN and ViT-VQGAN even with more model parameters, reaching the speed of efficient GAN models which only require 1 step to generate an image.

Table 2: **ImageNet 512×512 conditional generation.** †: quoted from MaskGIT [11]. "-s": a single shared AdaLN layer is used due to resource limitation.

Type	Model	FID↓	IS↑	Time
GAN	BigGAN [7]	8.43	177.9	–
Diff.	ADM [16]	23.24	101.0	–
Diff.	DiT-XL/2 [46]	3.04	240.8	81
Mask.	MaskGIT [11]	7.32	156.0	0.5
AR	VQGAN† [19]	26.52	66.8	25
VAR	VAR-$d36$-s	**2.63**	**303.2**	1

Compared with popular diffusion transformer.[5] The VAR model surpasses the recently popular diffusion models Diffusion Transformer (DiT), which serves as the precursor to the latest Stable-Diffusion 3 [18] and SORA [8], in multiple dimensions: 1) In image generation diversity and quality (FID and IS), VAR with 2B parameters consistently performs better than DiT-XL/2 [46], L-DiT-3B, and L-DiT-7B [2]. VAR also maintains comparable precision and recall. 2) For inference speed, the DiT-XL/2 requires 45× the wall-clock time compared to VAR, while 3B and 7B models [2] would cost much more. 3) VAR is considered more data-efficient, as it requires only 350 training epochs compared to DiT-XL/2's 1400. 4) For scalability, Fig. 3 and Tab. 1 show that DiT only obtains marginal or even negative gains beyond 675M parameters. In contrast, the FID and IS of VAR are consistently improved, aligning with the scaling law study in Sec. 4.2. These results establish *VAR as a more efficient and scalable model for image generation than models like DiT*.

4.2 Power-law scaling laws

Background. Prior research [30, 22, 27, 1] have established that scaling up autoregressive (AR) large language models (LLMs) leads to a predictable decrease in test loss L. This trend correlates with parameter counts N, training tokens T, and optimal training compute C_{min}, following a power-law:

$$L = (\beta \cdot X)^{\alpha}, \tag{9}$$

where X can be any of N, T, or C_{min}. The exponent α reflects the smoothness of power-law, and L denotes the reducible loss normalized by irreducible loss L_∞ [22][6]. A logarithmic transformation to L and X will reveal a linear relation between $\log(L)$ and $\log(X)$:

$$\log(L) = \alpha \log(X) + \alpha \log \beta. \tag{10}$$

These observed scaling laws [30, 22, 27, 1] not only validate the scalability of LLMs but also serve as a predictive tool for AR modeling, which facilitates the estimation of performance for larger AR models based on their smaller counterparts, thereby saving resource usage by large model performance forecasting. Given these appealing properties of scaling laws brought by LLMs, their replication in computer vision is therefore of significant interest.

Setup of scaling VAR models. Following the protocols from [30, 22, 27, 1], we examine whether our VAR model complies with similar scaling laws. We trained models across 12 different sizes, from 18M to 2B parameters, on the ImageNet training set [14] containing 1.28M images (or 870B image tokens under our VQVAE) per epoch. For models of different sizes, training spanned 200 to 350 epochs, with a maximum number of tokens reaching 305 billion. Below we focus on the scaling laws with model parameters N and optimal training compute C_{min} given sufficient token count T.

Scaling laws with model parameters N. We first investigate the test loss trend as the VAR model size increases. The number of parameters $N(d) = 73728\, d^3$ for a VAR transformer with depth d is specified in (8). We varied d from 6 to 30, yielding 12 models with 18.5M to 2.0B parameters. We assessed the final test cross-entropy loss L and token prediction error rates Err on the ImageNet validation set of 50,000 images [14]. We computed L and Err for both the last scale (at the last next-scale autoregressive step), as well as the global average. The results are plotted in Fig. 5, where we

[5] [46] do not report DiT-L/2's performance with CFG so we use official code to reproduce it.

[6] See [22] for some theoretical explanation on scaling laws on negative-loglikelihood losses.

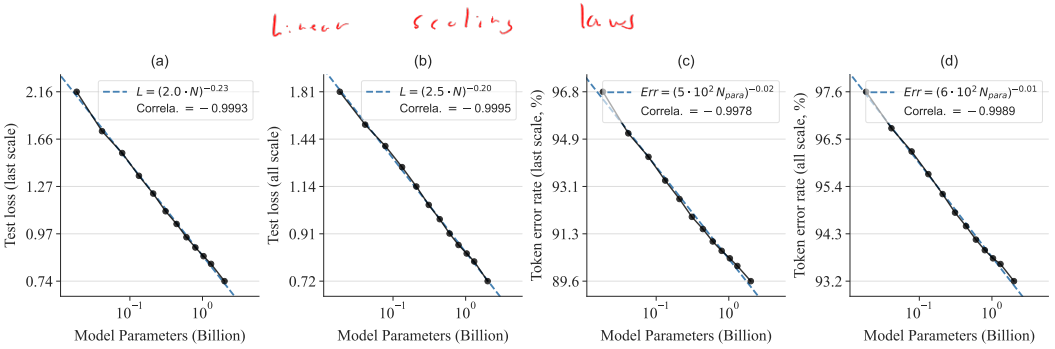


Figure 5: **Scaling laws with VAR transformer size** N, with power-law fits (dashed) and equations (in legend). Small, near-zero exponents α suggest a smooth decline in both test loss L and token error rate Err when scaling up VAR transformer. Axes are all on a logarithmic scale. The Pearson correlation coefficients near -0.998 signify a strong linear relationship between $log(N)$ *vs.* $log(L)$ or $log(N)$ *vs.* $log(Err)$.

observed a clear power-law scaling trend for L as a function of N, as consistent with [30, 22, 27, 1]. The power-law scaling laws can be expressed as:

$$L_{\text{last}} = (2.0 \cdot N)^{-0.23} \quad \text{and} \quad L_{\text{avg}} = (2.5 \cdot N)^{-0.20}. \tag{11}$$

Although the scaling laws are mainly studied on the test loss, we also empirically observed similar power-law trends for the token error rate Err:

$$Err_{\text{last}} = (4.9 \cdot 10^2 N)^{-0.016} \quad \text{and} \quad Err_{\text{avg}} = (6.5 \cdot 10^2 N)^{-0.010}. \tag{12}$$

These results verify the strong scalability of VAR, by which scaling up VAR transformers can continuously improve the model's test performance.

Scaling laws with optimal training compute C_{min}. We then examine the scaling behavior of VAR transformers when increasing training compute C. For each of the 12 models, we traced the test loss L and token error rate Err as a function of C during training quoted in PFlops (10^{15} floating-point operations per second). The results are plotted in Fig. 6. Here, we draw the Pareto frontier of L and Err to highlight the optimal training compute C_{min} required to reach a certain value of loss or error.

The fitted power-law scaling laws for L and Err as a function of C_{min} are:

$$L_{\text{last}} = (2.2 \cdot 10^{-5} C_{\text{min}})^{-0.13} \quad \text{and } L_{\text{avg}} = (1.5 \cdot 10^{-5} C_{\text{min}})^{-0.16}, \tag{13}$$

$$Err_{\text{last}} = (8.1 \cdot 10^{-2} C_{\text{min}})^{-0.0067} \quad \text{and } Err_{\text{avg}} = (4.4 \cdot 10^{-2} C_{\text{min}})^{-0.011}. \tag{14}$$

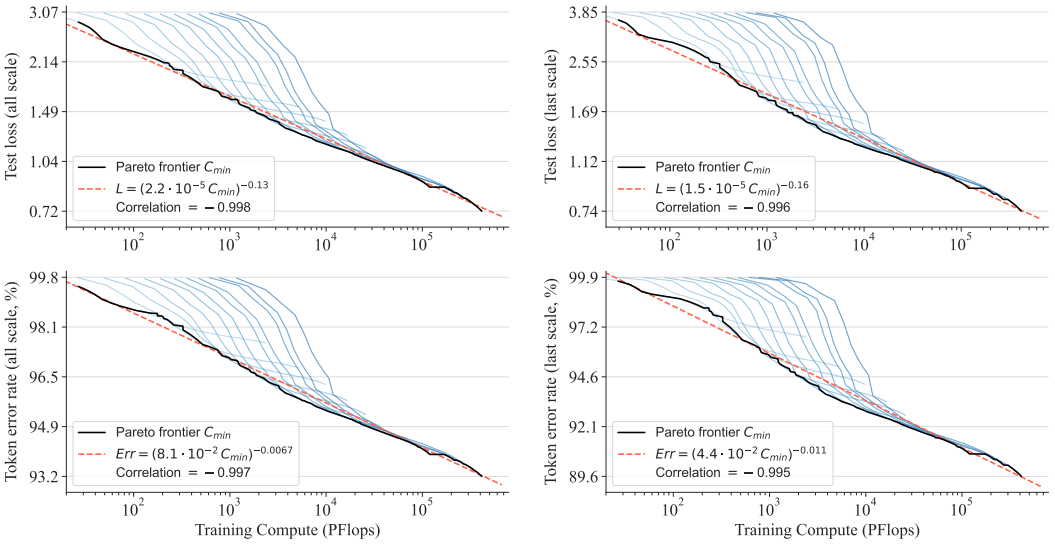


Figure 6: **Scaling laws with optimal training compute C_{min}.** Line color denotes different model sizes. Red dashed lines are power-law fits with equations in legend. Axes are on a logarithmic scale. Pearson coefficients near -0.99 indicate strong linear relationships between $\log(C_{\text{min}})$ *vs.* $\log(L)$ or $\log(C_{\text{min}})$ *vs.* $\log(Err)$.

These relations (13, 14) hold across 6 orders of magnitude in C_{min}, and our findings are consistent with those in [30, 22]: when trained with sufficient data, larger VAR transformers are more compute-efficient because they can reach the same level of performance with less computation.

Visualizations. To better understand how VAR models are learning when scaled up, we compare some generated 256×256 samples from VAR models of 4 different sizes (depth 6, 16, 26, 30) and 3 different training stages (20%, 60%, 100% of total training tokens) in Fig. 7. To keep the content consistent, a same random seed and teacher-forced initial tokens are used. The observed improvements in visual fidelity and soundness are consistent with the scaling laws, as larger transformers are thought able to learn more complex and fine-grained image distributions.

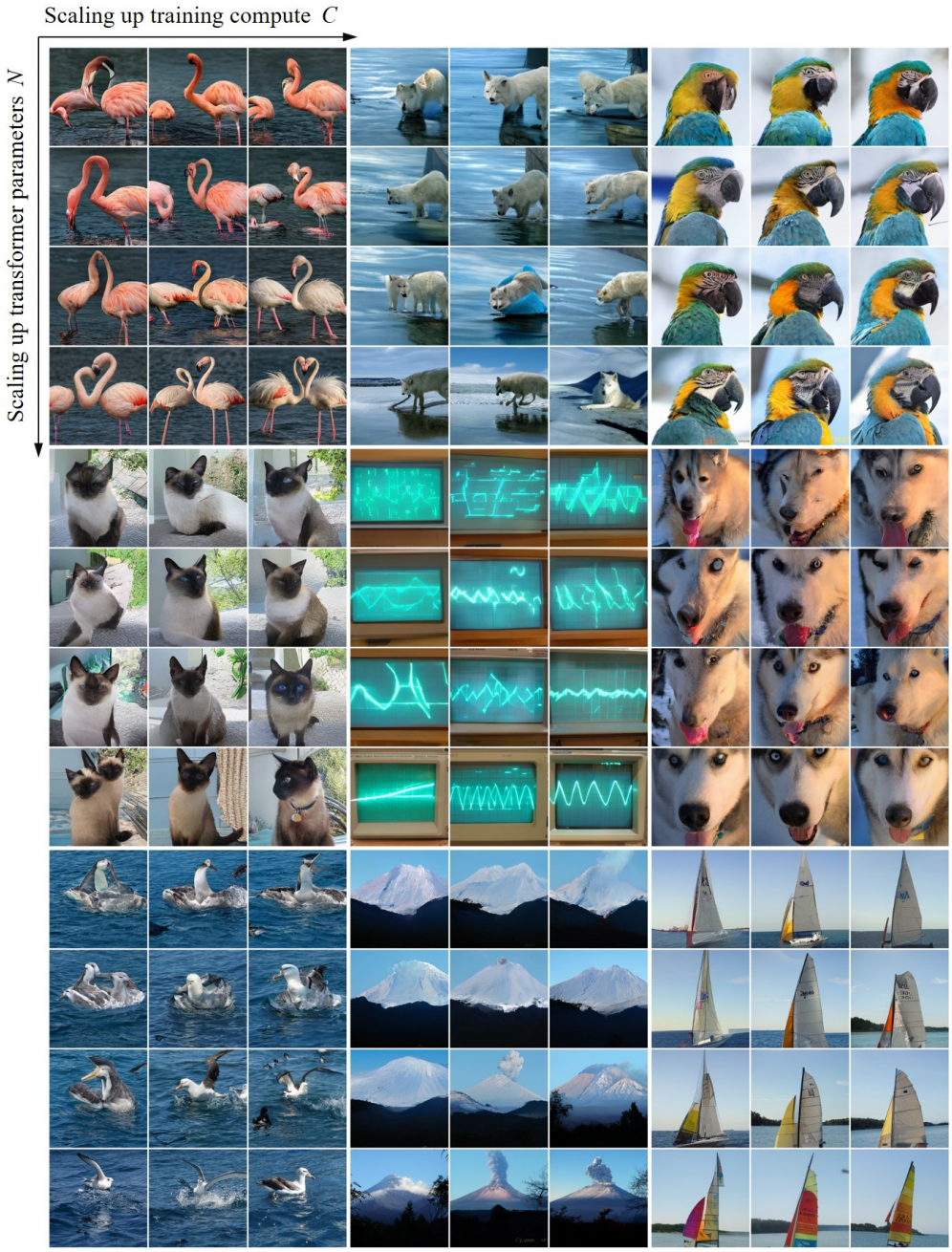


Figure 7: **Scaling model size N and training compute C improves visual fidelity and soundness.** Zoom in for a better view. Samples are drawn from VAR models of 4 different sizes and 3 different training stages. 9 class labels (from left to right, top to bottom) are: flamingo 130, arctic wolf 270, macaw 88, Siamese cat 284, oscilloscope 688, husky 250, mollymawk 146, volcano 980, and catamaran 484.

4.3 Zero-shot task generalization

Image in-painting and out-painting. VAR-$d30$ is tested. For in- and out-painting, we teacher-force ground truth tokens outside the mask and let the model only generate tokens within the mask. No class label information is injected into the model. The results are visualized in Fig. 8. Without modifications to the network architecture or tuning parameters, VAR has achieved decent results on these downstream tasks, substantiating the generalization ability of VAR.

Class-conditional image editing. Following MaskGIT [11] we also tested VAR on the class-conditional image editing task. Similar to the case of in-painting, the model is forced to generate tokens only in the bounding box conditional on some class label. Fig. 8 shows the model can produce plausible content that fuses well into the surrounding contexts, again verifying the generality of VAR.

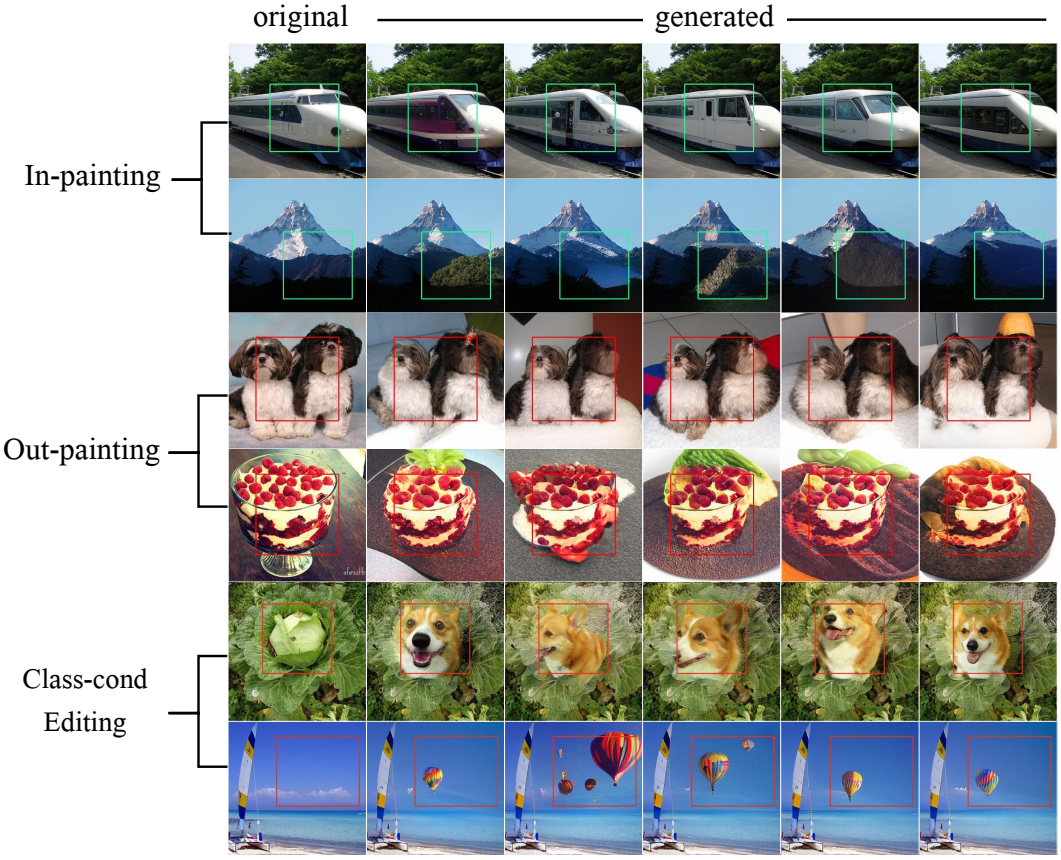


Figure 8: **Zero-shot evaluation in downstream tasks** containing in-painting, out-painting, and class-conditional editing. The results show that VAR can generalize to novel downstream tasks without special design and finetuning. Zoom in for a better view.

4.4 Ablation Study

In this study, we aim to verify the effectiveness and efficiency of our proposed VAR framework. Results are reported in Tab. 3.

Effectiveness and efficiency of VAR. Starting from the vanilla AR transformer baseline implemented by [11], we replace its methodology with our VAR and keep other settings unchanged to get row 2. VAR achieves a way more better FID (18.65 *vs.* 5.22) with only 0.013$\times$ inference wall-clock cost than the AR model, which demonstrates a leap in visual AR model's performance and efficiency.

Component-wise ablation. We further test some key components in VAR. By replacing the standard Layer Normalization (LN) with Adaptive Layer Normalization (AdaLN), VAR starts yielding better FID than baseline. By using the top-k sampling similar to the baseline, VAR's FID is further improved. By using the classifier-free guidance (CFG) with ratio 2.0, we reach the FID of 3.60, which is 15.05 lower to the baseline, and its inference is still 45 times faster. Due to the observed effectiveness, we equip our final VAR models with AdaLN, top-k sampling, and classifier-free guidance. We finally scale up VAR size to 2.0B and achieve an FID of 1.80. This is 16.85 better than the baseline FID.

Table 3: **Ablation study of VAR.** The first two rows compare GPT-2-style transformers trained with AR or VAR algorithm. Subsequent lines show the influence of VAR enhancements. "AdaLN": adaptive layernorm. "CFG": classifier-free guidance. "Cost": inference cost relative to the baseline. "Δ": reduction in FID.

	Description	Para.	Model	AdaLN	Top-k	CFG	Cost	FID↓	Δ
1	AR [11]	227M	AR	✗	✗	✗	1	18.65	0.00
2	AR to VAR	207M	VAR-$d16$	✗	✗	✗	0.013	5.22	−13.43
3	+AdaLN	310M	VAR-$d16$	✓	✗	✗	0.016	4.95	−13.70
4	+Top-k	310M	VAR-$d16$	✓	600	✗	0.016	4.64	−14.01
5	+CFG	310M	VAR-$d16$	✓	600	2.0	0.022	3.60	−15.05
6	+Scale up	2.0B	VAR-$d30$	✓	600	2.0	0.052	1.80	−16.85

5 Future Work

In this work, we mainly focus on the design of learning paradigm and keep the VQVAE architecture and training unchanged from the baseline [19] to better justify VAR framework's effectiveness. We expect **advancing VQVAE tokenizer** [77, 43, 74] as another promising way to enhance autoregressive generative models, which is orthogonal to our work. We believe iterating VAR by advanced tokenizer or sampling techniques in these latest work can further improve VAR's performance or speed.

Text-prompt generation is an ongoing direction of our research. Given that our model is fundamentally similar to modern LLMs, it can easily be integrated with them to perform text-to-image generation through either an encoder-decoder or in-context manner. This is currently in our high priority for exploration.

Video generation is not implemented in this work, but it can be naturally extended. By considering multi-scale video features as **3D pyramids**, we can formulate a similar "**3D next-scale prediction**" to generate videos via VAR. Compared to diffusion-based generators like SORA [8], our method has inherent advantages in temporal consistency or integration with LLMs, thus can potentially handle longer temporal dependencies. This makes VAR competitive in the video generation field, because traditional AR models can be too inefficient for video generation due to their extremely high computational complexity and slow inference speed: it is becoming prohibitively expensive to generate high-resolution videos with traditional AR models, while VAR is capable to solve this. We therefore foresee a promising future for exploiting VAR models in the realm of video generation.

6 Conclusion

We introduced a new visual generative framework named Visual AutoRegressive modeling (VAR) that 1) theoretically addresses some issues inherent in standard image autoregressive (AR) models, and 2) makes language-model-based AR models first surpass strong diffusion models in terms of image quality, diversity, data efficiency, and inference speed. Upon scaling VAR to 2 billion parameters, we observed a clear power-law relationship between test performance and model parameters or training compute, with Pearson coefficients nearing -0.998, indicating a robust framework for performance prediction. These scaling laws and the possibility for zero-shot task generalization, as hallmarks of LLMs, have now been initially verified in our VAR transformer models. We hope our findings and open sources can facilitate a more seamless integration of the substantial successes from the natural language processing domain into computer vision, ultimately contributing to the advancement of powerful multi-modal intelligence.

A Token dependency in VQVAE

To examine the token dependency in VQVAE [19], we check the attention scores in the self-attention layer before the vector quantization module. We randomly sample 4 256×256 images from the ImageNet validation set for this analysis. Note the self-attention layer in [19] only has 1 head so for each image we just plot one attention map. The heat map in Fig. 9 shows the attention scores of each token to all other tokens, which indicate a strong, bidirectional dependency among all tokens. This is not surprising since the VQVAE model, trained to reconstruct images, leverages self-attention layers without any attention mask. Some work [67] has used causal attention in self-attention layers of a video VAE, but we did not find any image VAE work uses causal self-attention.

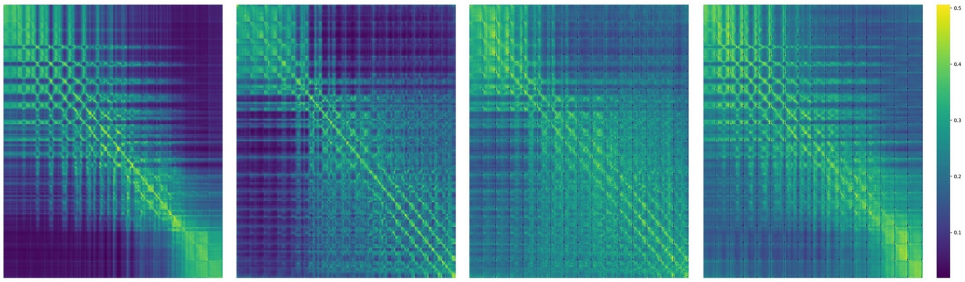


Figure 9: **Token dependency plotted.** The normalized heat map of attention scores in the last self-attention layer of VQGAN encoder is visualized. 4 random 256×256 images from ImageNet validation set are used.

B Time complexity of AR and VAR generation

We prove the time complexity of AR and VAR generation.

Lemma B.1. *For a standard self-attention transformer, the time complexity of AR generation is $\mathcal{O}(n^6)$, where $h = w = n$ and h, w are the height and width of the VQ code map, respectively.*

Proof. The total number of tokens is $h \times w = n^2$. For the i-th ($1 \le i \le n^2$) autoregressive iteration, the attention scores between each token and all other tokens need to be computed, which requires $\mathcal{O}(i^2)$ time. So the total time complexity would be:

$$\sum_{i=1}^{n^2} i^2 = \frac{1}{6} n^2 (n^2 + 1)(2n^2 + 1), \tag{15}$$

Which is equivalent to $\mathcal{O}(n^6)$ basic computation. □

For VAR, it needs us to define the resolution sequense $(h_1, w_1, h_2, w_2, \ldots, h_K, w_K)$ for autoregressive generation, where h_i, w_i are the height and width of the VQ code map at the i-th autoregressive step, and $h_K = h, w_K = w$ reaches the final resolution. Suppose $n_k = h_k = w_k$ for all $1 \le k \le K$ and $n = h = w$, for simplicity. We set the resolutions as $n_k = a^{(k-1)}$ where $a > 1$ is a constant such that $a^{(K-1)} = n$.

Lemma B.2. *For a standard self-attention transformer and given hyperparameter $a > 1$, the time complexity of VAR generation is $\mathcal{O}(n^4)$, where $h = w = n$ and h, w are the height and width of the last (largest) VQ code map, respectively.*

Proof. Consider the k-th ($1 \le k \le K$) autoregressive generation. The total number of tokens of current all token maps $(r_1, r_2, \ldots, r_k)$ is:

$$\sum_{i=1}^{k} n_i^2 = \sum_{i=1}^{k} a^{2\cdot(k-1)} = \frac{a^{2k} - 1}{a^2 - 1}. \tag{16}$$

So the time complexity of the k-th autoregressive generation would be:

$$\left(\frac{a^{2k} - 1}{a^2 - 1} \right)^2. \tag{17}$$

By summing up all autoregressive generations, we have:

$$\sum_{k=1}^{\log_a(n)+1} \left(\frac{a^{2k} - 1}{a^2 - 1} \right)^2 \tag{18}$$

$$= \frac{(a^4 - 1)\log n + \left(a^8 n^4 - 2a^6 n^2 - 2a^4(n^2 - 1) + 2a^2 - 1\right)\log a}{(a^2 - 1)^3 (a^2 + 1)\log a} \tag{19}$$

$$\sim \mathcal{O}(n^4). \tag{20}$$

This completes the proof. □

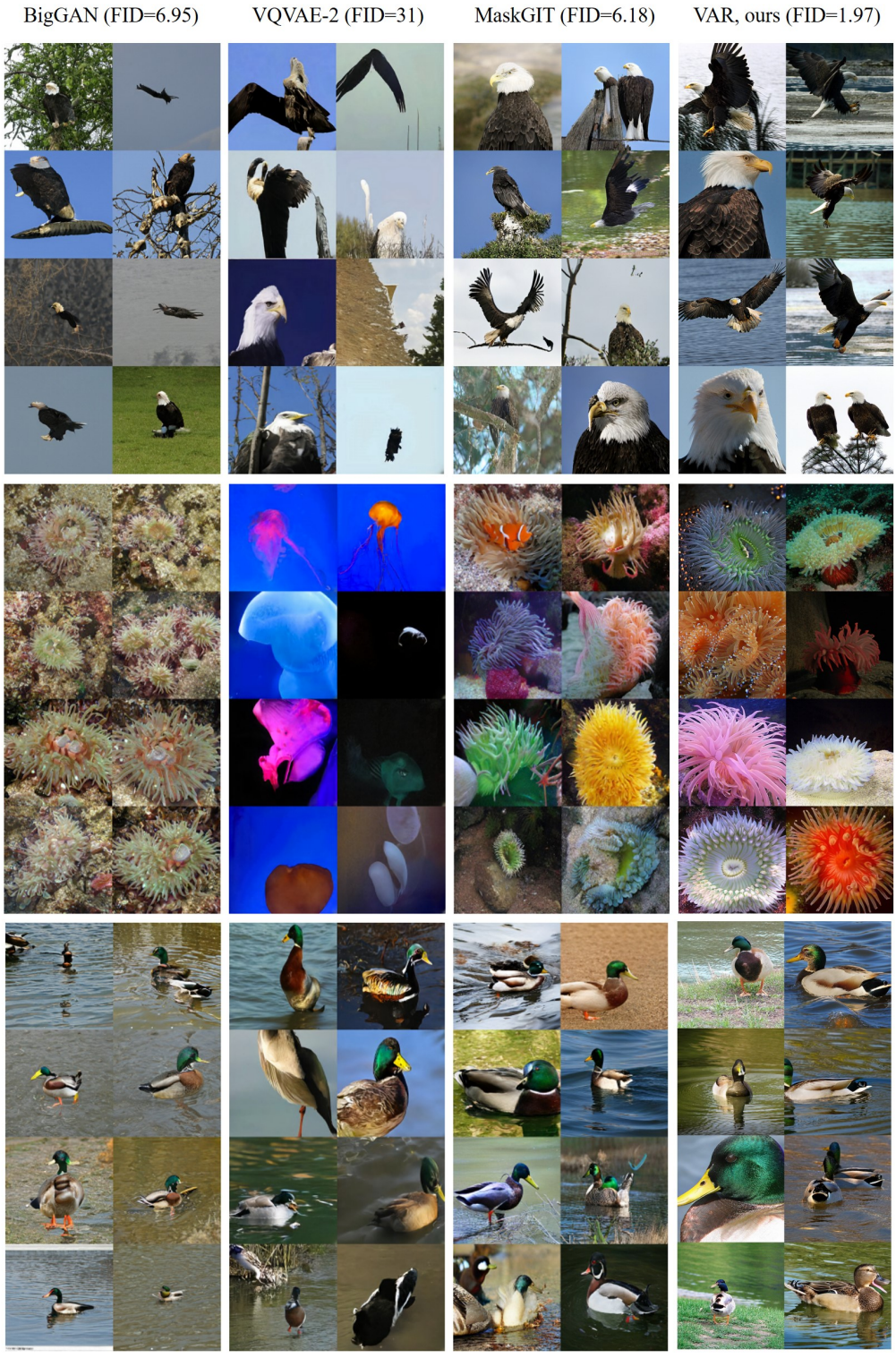


Figure 10: **Model comparison on ImageNet 256×256 benchmark.**

Figure 11: **More ImageNet 256×256 generation samples.**

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2, 8, 9

[2] Alpha-VLLM. Large-dit-imagenet. https://github.com/Alpha-VLLM/LLaMA2-Accessory/tree/f7fe19834b23e38f333403b91bb0330afe19f79e/Large-DiT-ImageNet, 2024. 2, 7, 8

[3] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023. 2

[4] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 2

[5] Y. Bai, X. Geng, K. Mangalam, A. Bar, A. Yuille, T. Darrell, J. Malik, and A. A. Efros. Sequential modeling enables scalable learning for large vision models. *arXiv preprint arXiv:2312.00785*, 2023. 2, 3

[6] H. Bao, L. Dong, S. Piao, and F. Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 4

[7] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 7, 8

[8] T. Brooks, B. Peebles, C. Holmes, W. DePue, Y. Guo, L. Jing, D. Schnurr, J. Taylor, T. Luhman, E. Luhman, C. Ng, R. Wang, and A. Ramesh. Video generation models as world simulators. *OpenAI*, 2024. 3, 4, 8, 12

[9] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 2, 3

[10] H. Chang, H. Zhang, J. Barber, A. Maschinot, J. Lezama, L. Jiang, M.-H. Yang, K. Murphy, W. T. Freeman, M. Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023. 4

[11] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022. 4, 7, 8, 11, 12

[12] J. Chen, J. Yu, C. Ge, L. Yao, E. Xie, Y. Wu, Z. Wang, J. Kwok, P. Luo, H. Lu, et al. Pixart: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023. 6

[13] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023. 2

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 8

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3, 4

[16] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 4, 7, 8

[17] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 4

[18] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, D. Podell, T. Dockhorn, Z. English, K. Lacey, A. Goodwin, Y. Marek, and R. Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024. 3, 4, 8

[19] P. Esser, R. Rombach, and B. 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, 3, 4, 5, 6, 7, 8, 12

[20] P. Gage. A new algorithm for data compression. *C Users Journal*, 12(2):23–38, 1994. 3

[21] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 4

[22] T. Henighan, J. Kaplan, M. Katz, M. Chen, C. Hesse, J. Jackson, H. Jun, T. B. Brown, P. Dhariwal, S. Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020. 2, 3, 8, 9, 10

[23] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 4

[24] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research*, 23(1):2249–2281, 2022. 4

[25] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research*, 23(1):2249–2281, 2022. 7

[26] J. Ho and T. Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 4

[27] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. 2, 3, 8, 9

[28] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*, 2023. 2

[29] M. Kang, J.-Y. Zhu, R. Zhang, J. Park, E. Shechtman, S. Paris, and T. Park. Scaling up gans for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10124–10134, 2023. 6, 7

[30] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 2, 3, 6, 8, 9, 10

[31] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 2

[32] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021. 6

[33] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 2, 4, 6

[34] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 6

[35] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 3

[36] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128(7):1956–1981, 2020. 6, 7

[37] D. Lee, C. Kim, S. Kim, M. Cho, and W.-S. 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, 3, 4, 6, 7, 8

[38] T. Li, D. Katabi, and K. He. Self-conditioned image generation via generating representations. *arXiv preprint arXiv:2312.03701*, 2023. 2, 7

[39] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 2

[40] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999. 2

[41] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022. 4

[42] J. Lu, C. Clark, R. Zellers, R. Mottaghi, and A. Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. *arXiv preprint arXiv:2206.08916*, 2022. 2

[43] F. Mentzer, D. Minnen, E. Agustsson, and M. Tschannen. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505*, 2023. 12

[44] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 3

[45] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022. 2

[46] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 2, 4, 6, 7, 8

[47] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3

[48] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. *article*, 2018. 2

[49] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 2, 3, 6

[50] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 4

[51] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 2

[52] A. Razavi, A. Van den Oord, and O. Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019. 2, 3, 7, 8

[53] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. 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. 4, 7

[54] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 4

[55] V. Sanh, A. Webson, C. Raffel, S. H. Bach, L. Sutawika, Z. Alyafeai, A. Chaffin, A. Stiegler, T. L. Scao, A. Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021. 3

[56] A. Sauer, T. Karras, S. Laine, A. Geiger, and T. Aila. Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis. *arXiv preprint arXiv:2301.09515*, 2023. 6

[57] A. Sauer, K. Schwarz, and A. Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022. 6, 7

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

[59] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019. 4

[60] Y. Sun, S. Wang, S. Feng, S. Ding, C. Pang, J. Shang, J. Liu, X. Chen, Y. Zhao, Y. Lu, et al. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*, 2021. 2

[61] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 2

[62] K. Tian, Y. Jiang, Q. Diao, C. Lin, L. Wang, and Z. Yuan. Designing bert for convolutional networks: Sparse and hierarchical masked modeling. *arXiv preprint arXiv:2301.03580*, 2023. 2

[63] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2, 3, 6

[64] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 2, 3, 6

[65] A. Van Den Oord, O. Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 3

[66] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4

[67] R. Villegas, M. Babaeizadeh, P.-J. Kindermans, H. Moraldo, H. Zhang, M. T. Saffar, S. Castro, J. Kunze, and D. Erhan. Phenaki: Variable length video generation from open domain textual descriptions. In *International Conference on Learning Representations*, 2022. 12

[68] W. Wang, Z. Chen, X. Chen, J. Wu, X. Zhu, G. Zeng, P. Luo, T. Lu, J. Zhou, Y. Qiao, et al. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[69] X. Wang, W. Wang, Y. Cao, C. Shen, and T. Huang. Images speak in images: A generalist painter for in-context visual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6830–6839, 2023. 3

[70] B. Workshop, T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022. 2, 3

[71] J. Yu, X. Li, J. Y. Koh, H. Zhang, R. Pang, J. Qin, A. Ku, Y. Xu, J. Baldridge, and Y. Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021. 2, 3, 4, 7, 8

[72] J. Yu, Y. Xu, J. Y. Koh, T. Luong, G. Baid, Z. Wang, V. Vasudevan, A. Ku, Y. Yang, B. K. Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022. 3

[73] L. Yu, Y. Cheng, K. Sohn, J. Lezama, H. Zhang, H. Chang, A. G. Hauptmann, M.-H. Yang, Y. Hao, I. Essa, et al. Magvit: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10459–10469, 2023. 4

[74] L. Yu, J. Lezama, N. B. Gundavarapu, L. Versari, K. Sohn, D. Minnen, Y. Cheng, A. Gupta, X. Gu, A. G. Hauptmann, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023. 3, 4, 7, 12

[75] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 4

[76] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 3

[77] C. Zheng, T.-L. Vuong, J. Cai, and D. Phung. Movq: Modulating quantized vectors for high-fidelity image generation. *Advances in Neural Information Processing Systems*, 35:23412–23425, 2022. 2, 12