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**[1,3,*]

[1]Center for Data Science, Peking University   [2]Bytedance Inc.
[3]State Key Lab of General Artificial Intelligence, School of
Intelligence Science and Technology, Peking University
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 1.73, inception score (IS) from 80.4 to 350.2, 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.

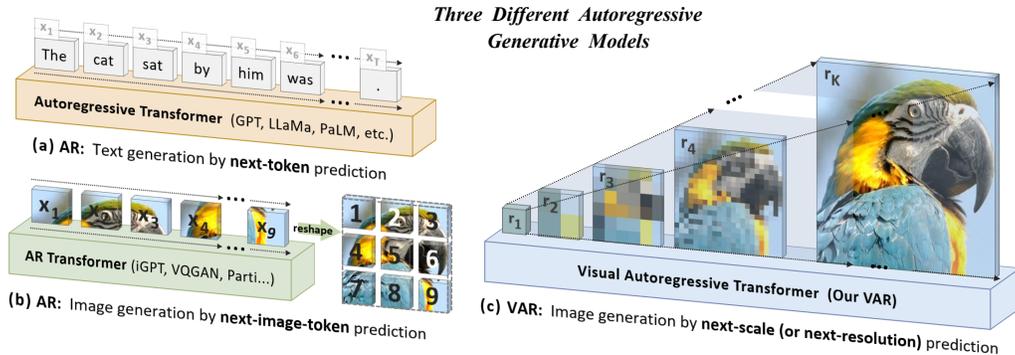*corresponding authors: wanglw@pku.edu.cn, yuanzehuan@bytedance.com; †: project lead

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 [66, 67, 15, 63, 1] and more autoregressive (AR) large language models (LLMs) [22, 4, 39, 83, 84, 91, 79, 5, 80] 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 [40], are still considered to take a solid step toward the general artificial intelligence (AGI). At the core of these 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* [44, 36], 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 [67, 15], 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 [59, 58, 6], aiming to emulate their impressive scalability and generalizability. Trailblazing efforts like VQGAN and DALL-E [30, 68] along with their successors [69, 92, 51, 99] 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 [64, 3, 52], 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 hierarchical manner, first capturing the global structure and then local details. This **multi-scale, coarse-to-fine** nature suggests an "order" for images. Also inspired by the widespread multi-scale designs [55, 53, 82, 45], 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.
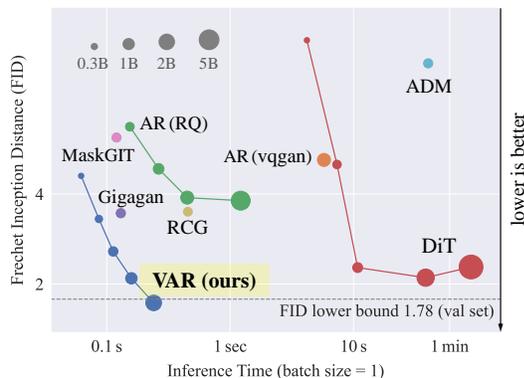

Figure 3: **Scaling behavior** of different model families on ImageNet 256×256 generation benchmark. The FID of the validation set serves as a reference lower bound (1.78). VAR with 2B parameters reaches an FID of 1.73, surpassing L-DiT with 3B or 7B parameters.

VAR directly leverages GPT-2-like transformer architecture [67] for visual autoregressive learning. On the ImageNet 256×256 benchmark, VAR significantly improves its AR baseline, achieving a Fréchet inception distance (FID) of 1.73 and an inception score (IS) of 350.2, with inference speed 20× faster (see Sec. 6 for details). Notably, VAR surpasses the Diffusion Transformer (DiT) – the foundation of leading diffusion systems like Stable Diffusion 3.0 and SORA [29, 14] – 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 are found and studied in *autoregressive* language models [44, 36], which describe a power-law relationship between the scale of model (or dataset, computation, *etc.*) and the cross-entropy loss value on the test set. Scaling laws allow us to directly predict the performance of a larger model from smaller ones [1], thus guiding better resource allocation. More pleasingly, they show that the performance of LLMs can scale well with the growth of model, data, and computation and never saturate, which is considered a key factor in the success of [15, 83, 84, 98, 91, 39]. The success brought by scaling laws has inspired the vision community to explore more similar methods for multimodality understanding and generation [54, 2, 89, 27, 96, 78, 21, 23, 42, 32, 33, 81, 88].

Zero-shot generalization. Zero-shot generalization [73] 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 computer vision, there is a burgeoning interest in the zero-shot and in-context learning abilities of foundation models, CLIP [65], SAM [49], Dinov2 [62]. Innovations like Painter [90] and LVM [6] extend visual prompters [41, 11] to achieve in-context learning in vision.

2.2 Visual generation

Raster-scan autoregressive models for visual generation necessitate the encoding of 2D images into 1D token sequences. Early endeavors [20, 85] have shown the ability to generate RGB (or grouped) pixels in the standard row-by-row, raster-scan manner. [70] extends [85] by using multiple independent trainable networks to do super-resolution repeatedly. VQGAN [30] advances [20, 85] by doing autoregressive learning in the latent space of VQVAE [86]. It employs GPT-2 decoder-only transformer to generate tokens in the raster-scan order, like how ViT [28] serializes 2D images into 1D patches. VQVAE-2 [69] and RQ-Transformer [51] also follow this raster-scan manner but use extra scales or stacked codes. Parti [93], based on the architecture of ViT-VQGAN [92], scales the transformer to 20B parameters and works well in text-to-image synthesis.

Masked-prediction model. MaskGIT [17] employs a VQ autoencoder and a masked prediction transformer similar to BERT [25, 10, 35] to generate VQ tokens through a greedy algorithm. MagViT [94] adapts this approach to videos, and MagViT-2 [95] enhances [17, 94] by introducing an improved VQVAE for both images and videos. MUSE [16] further scales MaskGIT to 3B parameters.

Diffusion models' progress has centered around improved learning or sampling [77, 76, 56, 57, 7], guidance [38, 61], latent learning [71], and architectures [37, 64, 72, 31]. DiT and U-ViT [64, 8] replaces or integrates the U-Net with transformer, and inspires recent image [19, 18] or video synthesis systems [12, 34] including Stable Diffusion 3.0 [29], SORA [14], and Vidu [9].

---

[2]A related work [95] named "language model beats diffusion" belongs to BERT-style masked-prediction model.

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 $x_t \in [V]$ is an integer from a vocabulary of size V. The next-token autoregressive posits the probability of observing the current token x_t depends only on its prefix $(x_1, x_2, \ldots, x_{t-1})$. This **unidirectional token dependency assumption** allows for the factorization of the sequence x's likelihood:

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

Training an autoregressive model p_θ involves optimizing $p_\theta(x_t \mid x_1, x_2, \ldots, x_{t-1})$ over a dataset. This 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 [30] 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:

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

$$\mathcal{L} = \|im - \hat{im}\|_2 + \|f - \hat{f}\|_2 + \lambda_\text{P} \mathcal{L}_\text{P}(\hat{im}) + \lambda_\text{G} \mathcal{L}_\text{G}(\hat{im}), \tag{5}$$

where $\mathcal{L}_\text{P}(\cdot)$ is a perceptual loss such as LPIPS [97], $\mathcal{L}_\text{G}(\cdot)$ a discriminative loss like StyleGAN's discriminator loss [47], 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 [30, 92, 51] 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.

Discussion on the weakness of vanilla autoregressive models. The above approach of tokenizing and flattening enable next-token autoregressive learning on images, but introduces several issues:

1) **Mathematical premise violation.** In quantized autoencoders (VQVAEs), the encoder typically produces an image feature map f with inter-dependent feature vectors $f^{(i,j)}$ for all i, j. So after quantization and flattening, the token sequence $(x_1, x_2, \ldots, x_{h \times w})$ retains *bidirectional* correlations. This contradicts the *unidirectional* dependency assumption of autoregressive models, which dictates that each token x_t should only depend on its prefix $(x_1, x_2, \ldots, x_{t-1})$.

2) **Inability to perform some zero-shot generalization.** Similar to issue 1), The unidirectional nature of image autoregressive modeling restricts their generalizability in tasks requiring bidirectional reasoning. E.g., it cannot predict the top part of an image given the bottom part.

3) **Structural degradation.** The flattening disrupts the spatial locality inherent in image feature maps. For example, 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.

4

4) **Inefficiency.** Generating an image token sequence $x = (x_1, x_2, \ldots, x_{n \times n})$ with a conventional self-attention transformer incurs $\mathcal{O}(n^2)$ autoregressive steps and $\mathcal{O}(n^6)$ computational cost.

Issues 2) and 3) are evident (see examples above). Regarding issue 1), we present empirical evidence in Appendix C. The proof of issue 3) is detailed in Appendix D. These theoretical and practical limitations call for a rethinking of autoregressive models in the context of image generation.
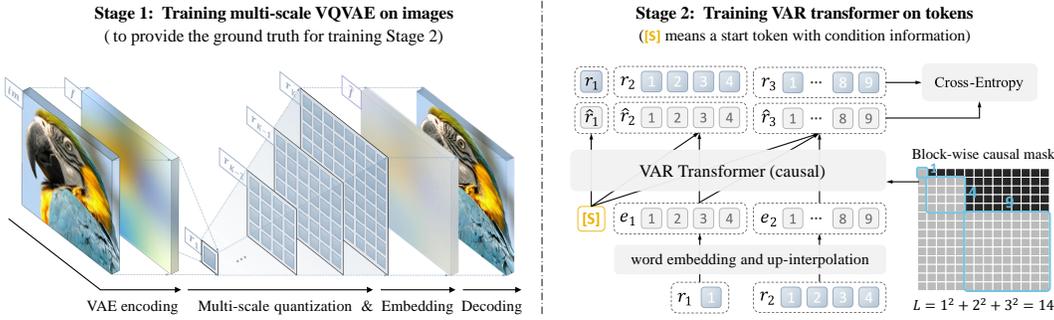

Stage 1: Training multi-scale VQVAE on images
(to provide the ground truth for training Stage 2)

Stage 2: Training VAR transformer on tokens
([S] means a start token with condition information)

VAE encoding   Multi-scale quantization & Embedding Decoding

VAR Transformer (causal)

word embedding and up-interpolation

Block-wise causal mask

$L = 1^2 + 2^2 + 3^2 = 14$

Figure 4: **VAR involves two separated training stages. Stage 1:** a multi-scale VQ autoencoder encodes an image into K token maps $R = (r_1, r_2, \ldots, r_K)$ and is trained by a compound loss (5). For details on "Multi-scale quantization" and "Embedding", check Algorithm 1 and 2. **Stage 2:** a VAR transformer is trained via next-scale prediction (6): it takes $([\texttt{s}], r_1, r_2, \ldots, r_{K-1})$ as input to predict $(r_1, r_2, r_3, \ldots, r_K)$. The attention mask is used in training to ensure each r_k can only attend to $r_{\leq k}$. Standard cross-entropy loss is used.

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 $(r_1, r_2, \ldots, r_K)$, 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:

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

where each autoregressive unit $r_k \in [V]^{h_k \times w_k}$ is the token map at scale k containing $h_k \times w_k$ tokens, and the sequence $(r_1, r_2, \ldots, r_{k-1})$ 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 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. Note that in the training of VAR, a block-wise causal attention mask is used to ensure that each r_k can only attend to its prefix $r_{\leq k}$. During inference, kv-caching can be used and no mask is needed.

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 $r_{\leq k}$. This constraint is acceptable as it aligns with the natural, coarse-to-fine progression characteristics like human visual perception and artistic drawing (as we discussed in Sec. 1). 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 develop 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 [30] 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 [51], can perform better than independent interpolation. Algorithm 1 shows that each r_k would depend only on its prefix $(r_1, r_2, \ldots, r_{k-1})$.

5

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

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

Algorithm 2: Multi-scale VQVAE Reconstruction

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

---

4 Implementation details

VAR tokenizer. As aforementioned, we use the vanilla VQVAE architecture [30] and a multi-scale quantization scheme with K extra convolutions (0.03M extra parameters). We use a shared codebook for all scales with $V = 4096$. Following the baseline [30], our tokenizer is also trained on OpenImages [50] with the compound loss (5) and a spatial downsample ratio of $16\times$.

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 VQ-GAN [67, 30] with adaptive normalization (AdaLN), which has widespread adoption and proven effectiveness in many visual generative models [47, 48, 46, 75, 74, 43, 64, 19]. For class-conditional synthesis, we use the class embedding as the start token [s] and also the condition of AdaLN. We found normalizing $queries$ and $keys$ to unit vectors before attention can stablize the training. We do not use advanced techniques in large language models, such as rotary position embedding (RoPE), SwiGLU MLP, or RMS Norm [83, 84]. Our model shape follows a simple rule like [44] 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 768 to 1024 and training epochs from 200 to 350 (depends on model size). The evaluations in Sec. 5 suggest that such a simple model design are capable of scaling and generalizing well.

5 Empirical Results

This section first compares VAR with other image generative model families in Sec. 5.1. Evaluations on the scalability and generalizability of VAR models are presented in Sec. 5.2 and Appendix B. For implementation details and ablation study, please see Appendix 4 and Appendix 6.

5.1 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 [30] and ours use the same architecture (CNN) and training data (OpenImages [50]) for VQVAE, while ViT-VQGAN [92] uses a ViT autoencoder, and both it and RQTransformer [51] trains the VQVAE directly on ImageNet. The results are summaried in Tab. 1 and Tab. 2.

---

[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 [17].

Type	Model	FID↓	IS↑	Pre↑	Rec↑	#Para	#Step	Time
GAN	BigGAN [13]	6.95	224.5	**0.89**	0.38	112M	1	–
GAN	GigaGAN [43]	3.45	225.5	0.84	**0.61**	569M	1	–
GAN	StyleGan-XL [75]	2.30	265.1	0.78	0.53	166M	1	0.3 [75]
Diff.	ADM [26]	10.94	101.0	0.69	0.63	554M	250	168 [75]
Diff.	CDM [37]	4.88	158.7	–	–	–	8100	–
Diff.	LDM-4-G [71]	3.60	247.7	–	–	400M	250	–
Diff.	DiT-L/2 [64]	5.02	167.2	0.75	0.57	458M	250	31
Diff.	DiT-XL/2 [64]	2.27	278.2	0.83	0.57	675M	250	45
Diff.	L-DiT-3B [3]	2.10	304.4	0.82	0.60	3.0B	250	>45
Diff.	L-DiT-7B [3]	2.28	316.2	0.83	0.58	7.0B	250	>45
Mask.	MaskGIT [17]	6.18	182.1	0.80	0.51	227M	8	0.5 [17]
Mask.	RCG (cond.) [52]	3.49	215.5	–	–	502M	20	1.9 [52]
AR	VQVAE-2† [69]	31.11	~45	0.36	0.57	13.5B	5120	–
AR	VQGAN† [30]	18.65	80.4	0.78	0.26	227M	256	19 [17]
AR	VQGAN [30]	15.78	74.3	–	–	1.4B	256	24
AR	VQGAN-re [30]	5.20	280.3	–	–	1.4B	256	24
AR	ViTVQ [92]	4.17	175.1	–	–	1.7B	1024	>24
AR	ViTVQ-re [92]	3.04	227.4	–	–	1.7B	1024	>24
AR	RQTran. [51]	7.55	134.0	–	–	3.8B	68	21
VAR	VAR-d16	3.30	274.4	0.84	0.51	310M	10	0.4
VAR	VAR-d20	2.57	302.6	0.83	0.56	600M	10	0.5
VAR	VAR-d24	2.09	312.9	0.82	0.59	1.0B	10	0.6
VAR	VAR-d30	1.92	323.1	0.82	0.59	2.0B	10	1
VAR	VAR-d30-re	**1.73**	**350.2**	0.82	0.60	2.0B	10	1
	(validation data)	1.78	236.9	0.75	0.67			

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.

Efficiency comparison. Conventional autoregressive (AR) models [30, 69, 92, 51] 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 [17]. "-s": a single shared AdaLN layer is used due to resource limitation.

Type	Model	FID↓	IS↑	Time
GAN	BigGAN [13]	8.43	177.9	–
Diff.	ADM [26]	23.24	101.0	–
Diff.	DiT-XL/2 [64]	3.04	240.8	81
Mask.	MaskGIT [17]	7.32	156.0	0.5†
AR	VQGAN [30]	26.52	66.8	25†
VAR	VAR-d36-s	**2.63**	**303.2**	1

Compared with popular diffusion transformer. The VAR model surpasses the recently popular diffusion models Diffusion Transformer (DiT), which serves as the precursor to the latest Stable-Diffusion 3 [29] and SORA [14], in multiple dimensions: 1) In image generation diversity and quality

(FID and IS), VAR with 2B parameters consistently performs better than DiT-XL/2 [64], L-DiT-3B, and L-DiT-7B [3]. 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 [3] 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. 5.2. These results establish *VAR as potentially a more efficient and scalable model for image generation than models like DiT*.
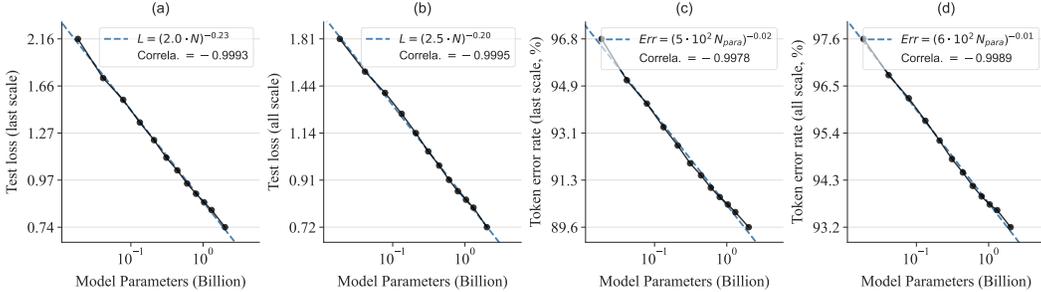

Figure 5: **Scaling laws with VAR transformer size** N, with power-law fits (dashed) and equations (in legend). Small, near-zero exponents α suggest a smooth decline in both test loss L and token error rate Err when scaling up VAR transformer. Axes are all on a logarithmic scale. The Pearson correlation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

5.2 Power-law scaling laws

Background. Prior research [44, 36, 39, 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_{∞} [36] A logarithmic transformation to L and X will reveal a linear relation between $\log(L)$ and $\log(X)$:

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

An appealing phenomenon is that both [44] and [36] never observed deviation from these linear relationships at the higher end of X, although flattening is inevitable as the loss approaches zero.

These observed scaling laws [44, 36, 39, 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 [44, 36, 39, 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 [24] 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 [24]. We computed L and Err for both the last scale (at the last next-scale autoregressive step), as well as the global average. Results are plotted in Fig. 5, where we
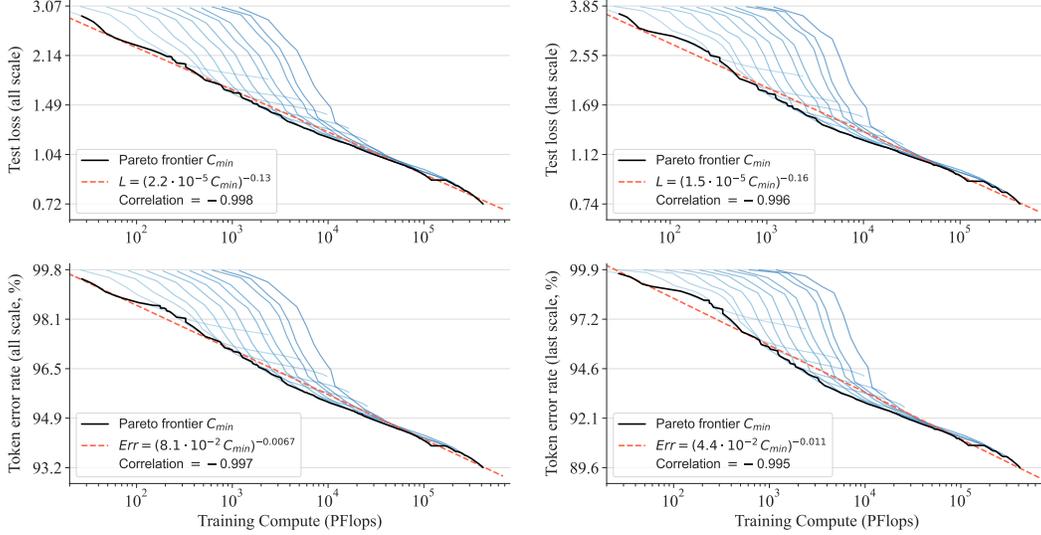
8

Figure 6: **Scaling laws with optimal training compute** $C_{\mathbf{min}}$**.** 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 $\log(C_{\min})$ *vs.* $\log(L)$ or $\log(C_{\min})$ *vs.* $\log(Err)$.

observed a clear power-law scaling trend for L as a function of N, as consistent with [44, 36, 39, 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 $C_{\mathbf{min}}$**.** 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:

$$L_{\text{last}} = (2.2 \cdot 10^{-5} C_{\min})^{-0.13} \tag{13}$$

$$L_{\text{avg}} = (1.5 \cdot 10^{-5} C_{\min})^{-0.16}, \tag{14}$$

$$Err_{\text{last}} = (8.1 \cdot 10^{-2} C_{\min})^{-0.0067} \tag{15}$$

$$Err_{\text{avg}} = (4.4 \cdot 10^{-2} C_{\min})^{-0.011}. \tag{16}$$

These relations (14, 16) hold across 6 orders of magnitude in $C_{\min}$, and our findings are consistent with those in [44, 36]: when trained with sufficient data, larger VAR transformers are more compute-efficient because they can reach the same level of performance with less computation.

6   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 [17], we replace its methodology with our VAR and keep other settings unchanged to get row 2.

Table 3: **Ablation study of VAR.** The first two rows compare GPT-2-style transformers trained under AR or VAR algorithm without any bells and whistles. Subsequent lines show the influence of VAR enhancements. "AdaLN": adaptive layernorm. "CFG": classifier-free guidance. "Attn. Norm.": normalizing q and k to unit vectors before attention. "Cost": inference cost relative to the baseline. "Δ": FID reduction to the baseline.

	Description	Para.	Model	AdaLN	Top-k	CFG	Cost	FID↓	Δ
1	AR [30]	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	✓	900	✗	0.016	4.64	−14.01
5	+CFG	310M	VAR-d16	✓	900	1.5	0.022	3.60	−15.05
5	+Attn. Norm.	310M	VAR-d16	✓	900	1.5	0.022	3.30	−15.35
6	+Scale up	2.0B	VAR-d30	✓	900	1.5	0.052	1.73	−16.85

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 1.5 and normalizing q and k to unit vectors before attention, we reach the FID of 3.30, which is 15.35 lower to the baseline, and its inference speed is still 45 times faster. We finally scale up VAR size to 2.0B and achieve an FID of 1.73. This is 16.85 better than the baseline FID.

7   Limitations and 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 [30] to better justify VAR framework's effectiveness. We expect **advancing VQVAE tokenizer** [99, 60, 95] 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.

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 [14], 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.

8   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.

9 Acknowledgements

A Visualization of scaling effect

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.

B 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 [17] 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.

C Token dependency in VQVAE

To examine the token dependency in VQVAE [30], 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 [30] 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 [87] 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.

D Time complexity of AR and VAR generation

We prove the time complexity of AR and VAR generation.

Lemma D.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 ($1 \leq i \leq n^2$) 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:

$$\sum_{i=1}^{n^2} i^2 = \frac{1}{6}n^2(n^2 + 1)(2n^2 + 1), \tag{17}$$

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

For VAR, it needs us to define the resolution sequence $(h_1, w_1, h_2, w_2, \ldots, h_K, w_K)$ 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 $1 \leq k \leq K$
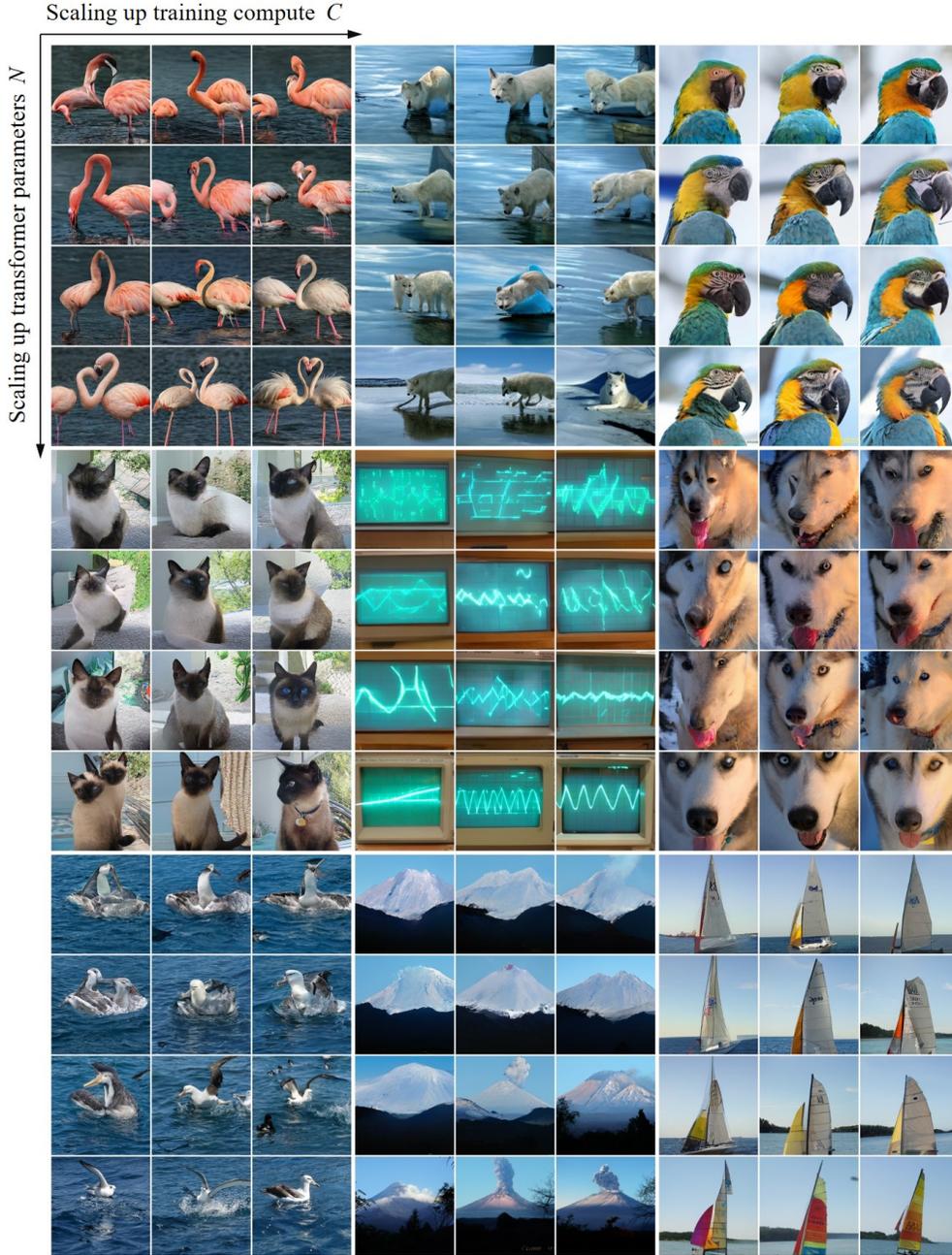
Figure 7: **Scaling model size N and training compute C improves visual fidelity and soundness.** Zoom in for a better view. Samples are drawn from VAR models of 4 different sizes and 3 different training stages. 9 class labels (from left to right, top to bottom) are: flamingo 130, arctic wolf 270, macaw 88, Siamese cat 284, oscilloscope 688, husky 250, mollymawk 146, volcano 980, and catamaran 484.

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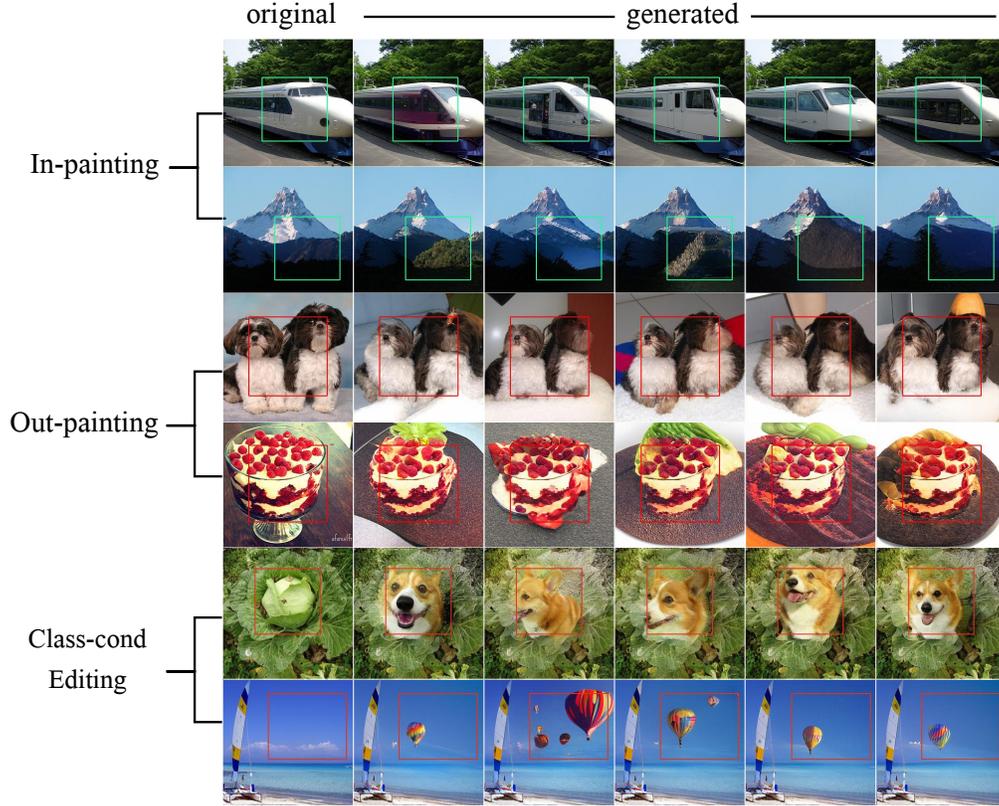
Figure 8: **Zero-shot evaluation in downstream tasks** containing in-painting, out-painting, and class-conditional editing. The results show that VAR can generalize to novel downstream tasks without special design and finetuning. Zoom in for a better view.
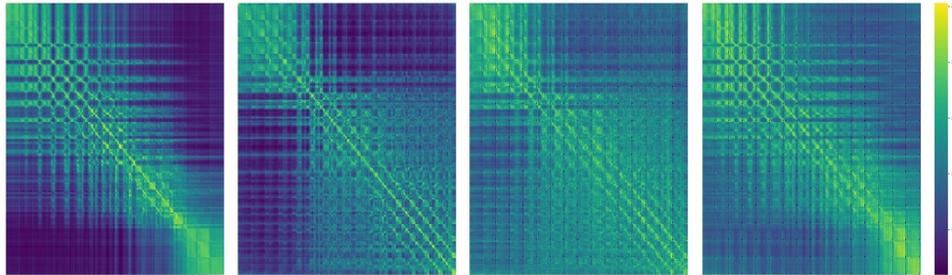

Figure 9: **Token dependency plotted.** The normalized heat map of attention scores in the last self-attention layer of VQGAN encoder is visualized. 4 random 256×256 images from ImageNet validation set are used.

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

$$\sum_{i=1}^{k} n_i^2 = \sum_{i=1}^{k} a^{2 \cdot (k-1)} = \frac{a^{2k} - 1}{a^2 - 1}. \tag{18}$$

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

$$\left(\frac{a^{2k} - 1}{a^2 - 1} \right)^2. \tag{19}$$

13

By summing up all autoregressive generations, we have:

$$\sum_{k=1}^{\log_a(n)+1} \left(\frac{a^{2k}-1}{a^2-1}\right)^2 \tag{20}$$

$$= \frac{(a^4-1)\log n + \left(a^8 n^4 - 2a^6 n^2 - 2a^4(n^2-1) + 2a^2 - 1\right)\log a}{(a^2-1)^3(a^2+1)\log a} \tag{21}$$

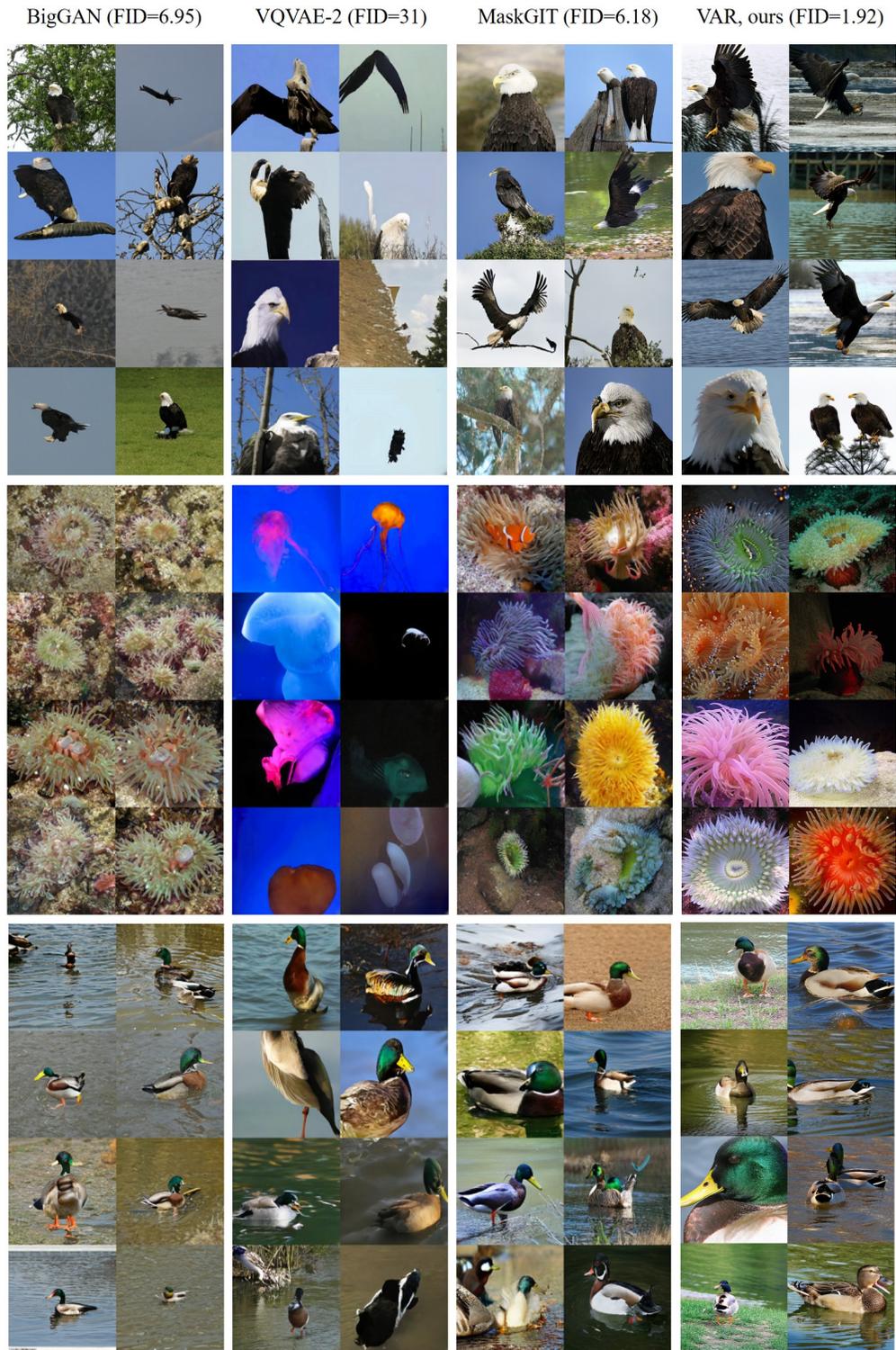$$\sim \mathcal{O}(n^4). \tag{22}$$

This completes the proof. $\qquad\qquad\square$

Figure 10: **Model comparison on ImageNet 256×256 benchmark.** More generated 512×512 samples by VAR can be found in the submitted Supplementary Material zip file.

Figure 11: **Some generated 256×256 samples by VAR trained on ImageNet.** More generated 512×512 samples by VAR can be found in the submitted Supplementary Material zip file.

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, 3, 8, 9

[2] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. 3

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

[4] 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

[5] 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

[6] 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

[7] F. Bao, C. Li, J. Zhu, and B. Zhang. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. *arXiv preprint arXiv:2201.06503*, 2022. 4

[8] F. Bao, S. Nie, K. Xue, Y. Cao, C. Li, H. Su, and J. Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22669–22679, 2023. 4

[9] F. Bao, C. Xiang, G. Yue, G. He, H. Zhu, K. Zheng, M. Zhao, S. Liu, Y. Wang, and J. Zhu. Vidu: a highly consistent, dynamic and skilled text-to-video generator with diffusion models. *arXiv preprint arXiv:2405.04233*, 2024. 4

[10] H. Bao, L. Dong, S. Piao, and F. Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 3

[11] A. Bar, Y. Gandelsman, T. Darrell, A. Globerson, and A. Efros. Visual prompting via image inpainting. *Advances in Neural Information Processing Systems*, 35:25005–25017, 2022. 3

[12] O. Bar-Tal, H. Chefer, O. Tov, C. Herrmann, R. Paiss, S. Zada, A. Ephrat, J. Hur, Y. Li, T. Michaeli, et al. Lumiere: A space-time diffusion model for video generation. *arXiv preprint arXiv:2401.12945*, 2024. 4

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

[14] 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, 10

[15] 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

[16] 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. 3

[17] 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. 3, 7, 10, 11

[18] J. Chen, C. Ge, E. Xie, Y. Wu, L. Yao, X. Ren, Z. Wang, P. Luo, H. Lu, and Z. Li. Pixart-\sigma: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. *arXiv preprint arXiv:2403.04692*, 2024. 4

[19] 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. 4, 6

[20] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020. 3

[21] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, Z. Muyan, Q. Zhang, X. Zhu, L. Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023. 3

[22] 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

[23] X. Dai, J. Hou, C.-Y. Ma, S. Tsai, J. Wang, R. Wang, P. Zhang, S. Vandenhende, X. Wang, A. Dubey, et al. Emu: Enhancing image generation models using photogenic needles in a haystack. *arXiv preprint arXiv:2309.15807*, 2023. 3

[24] 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, 9, 22, 23

[25] 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

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

[27] R. Dong, C. Han, Y. Peng, Z. Qi, Z. Ge, J. Yang, L. Zhao, J. Sun, H. Zhou, H. Wei, et al. Dreamllm: Synergistic multimodal comprehension and creation. *arXiv preprint arXiv:2309.11499*, 2023. 3

[28] 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. 3

[29] 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

[30] 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, 6, 7, 10, 11

[31] S. Gao, P. Zhou, M.-M. Cheng, and S. Yan. Mdtv2: Masked diffusion transformer is a strong image synthesizer. *arXiv preprint arXiv:2303.14389*, 2023. 4

[32] Y. Ge, S. Zhao, Z. Zeng, Y. Ge, C. Li, X. Wang, and Y. Shan. Making llama see and draw with seed tokenizer. *arXiv preprint arXiv:2310.01218*, 2023. 3

[33] Y. Ge, S. Zhao, J. Zhu, Y. Ge, K. Yi, L. Song, C. Li, X. Ding, and Y. Shan. Seed-x: Multimodal models with unified multi-granularity comprehension and generation. *arXiv preprint arXiv:2404.14396*, 2024. 3

[34] A. Gupta, L. Yu, K. Sohn, X. Gu, M. Hahn, L. Fei-Fei, I. Essa, L. Jiang, and J. Lezama. Photorealistic video generation with diffusion models. *arXiv preprint arXiv:2312.06662*, 2023. 4

[35] 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. 3

[36] 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

[37] 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, 7

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

[39] 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

[40] 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

[41] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, and S.-N. Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727. Springer, 2022. 3

[42] Y. Jin, K. Xu, L. Chen, C. Liao, J. Tan, B. Chen, C. Lei, A. Liu, C. Song, X. Lei, et al. Unified language-vision pretraining with dynamic discrete visual tokenization. *arXiv preprint arXiv:2309.04669*, 2023. 3

[43] 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

[44] 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

[45] 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

[46] 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

[47] 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. 4, 6

[48] 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

[49] 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

[50] 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

[51] 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

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

[53] 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

[54] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 3

[55] 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

[56] 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

[57] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022. 4

[58] J. Lu, C. Clark, S. Lee, Z. Zhang, S. Khosla, R. Marten, D. Hoiem, and A. Kembhavi. Unified-io 2: Scaling autoregressive multimodal models with vision, language, audio, and action. *arXiv preprint arXiv:2312.17172*, 2023. 2

[59] 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

[60] F. Mentzer, D. Minnen, E. Agustsson, and M. Tschannen. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505*, 2023. 10

[61] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 4

[62] 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

[63] 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

[64] 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

[65] 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

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

[67] 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

[68] 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

[69] 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

[70] S. Reed, A. Oord, N. Kalchbrenner, S. G. Colmenarejo, Z. Wang, Y. Chen, D. Belov, and N. Freitas. Parallel multiscale autoregressive density estimation. In *International conference on machine learning*, pages 2912–2921. PMLR, 2017. 3

[71] 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

[72] 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

[73] 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

[74] 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

[75] 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

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

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

[78] Q. Sun, Q. Yu, Y. Cui, F. Zhang, X. Zhang, Y. Wang, H. Gao, J. Liu, T. Huang, and X. Wang. Generative pretraining in multimodality. *arXiv preprint arXiv:2307.05222*, 2023. 3

[79] 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

[80] 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

[81] C. Tian, X. Zhu, Y. Xiong, W. Wang, Z. Chen, W. Wang, Y. Chen, L. Lu, T. Lu, J. Zhou, et al. Mm-interleaved: Interleaved image-text generative modeling via multi-modal feature synchronizer. *arXiv preprint arXiv:2401.10208*, 2024. 3

[82] 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

[83] 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

[84] 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

[85] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016. 3

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

[87] 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. 11

[88] H. Wang, H. Tang, L. Jiang, S. Shi, M. F. Naeem, H. Li, B. Schiele, and L. Wang. Git: Towards generalist vision transformer through universal language interface. *arXiv preprint arXiv:2403.09394*, 2024. 3

[89] 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. 3

[90] 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

[91] 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

[92] 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

[93] 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

[94] 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. 3

[95] 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, 10

[96] L. Yu, B. Shi, R. Pasunuru, B. Muller, O. Golovneva, T. Wang, A. Babu, B. Tang, B. Karrer, S. Sheynin, et al. Scaling autoregressive multi-modal models: Pretraining and instruction tuning. *arXiv preprint arXiv:2309.02591*, 2(3), 2023. 3

[97] 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

[98] 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

[99] 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, 10

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Yes. Our main contributions are also detailed in Sec. 1. Also see Sec. 5 and Appendix D for more theoretical and experimental evidence.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Yes, please see Sec. 7 for limitations. We also reported a lot about computational efficiency, such as in Tab. 1 and Appendix D.

   Guidelines:

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

3. **Theory Assumptions and Proofs**

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

Answer: [Yes]

Justification: We detail the assumption and proof of theoretical result on time complexity in Appendix D.

Guidelines:

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

4. **Experimental Result Reproducibility**

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

   Answer: [Yes]

   Justification: We use publicly-accessable dataset ImageNet [24]. We upload the codes and instructions to recover the results.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We use publicly-accessible dataset ImageNet [24]. We upload the codes and instructions to recover the results. Once the blind review period is finished, we'll open-source all codes, instructions, and model checkpoints.

Guidelines:

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

6. **Experimental Setting/Details**

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

Answer: [Yes]

Justification: Please see Sec. 5 and Appendix 4.

Guidelines:

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

7. **Experiment Statistical Significance**

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

Answer: [No]

Justification: Due to the resource limitation, we do not report error bars. Please note that in Sec. 5 we spent numerous resources (we trained 12 different models) for our scaling law study, which makes it prohibitively to run each experiments for multiple times.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments Compute Resources**

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

Answer: [Yes]

Justification: We report the training PFlops in Fig. 6 and speed in Tab. 1 and Tab. 2.

Guidelines:

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

9. **Code Of Ethics**

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

Answer: [Yes]

Justification: We followed the NeurIPS Code of Ethics.

Guidelines:

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

10. **Broader Impacts**

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

Answer: [No]

Justification: This work focuses on a academic, publicly-available benchmark ImageNet. This work is not related to any private or personal data, and there's no explicit negative social impacts.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.

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

11. **Safeguards**

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

Answer: [No]

Justification: We do not foresee any high risk for misuse of this work.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: Yes, we credited them in appropriate ways.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

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

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

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

14. **Crowdsourcing and Research with Human Subjects**

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

Answer: [NA]

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

Guidelines:

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

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

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

Answer: [NA]

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

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.