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

NeurIPS 2024 Best Paper

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

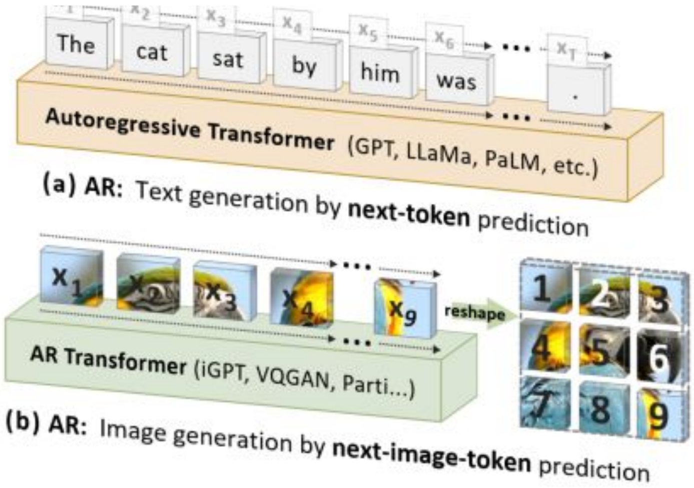
2025.04.15
채진영

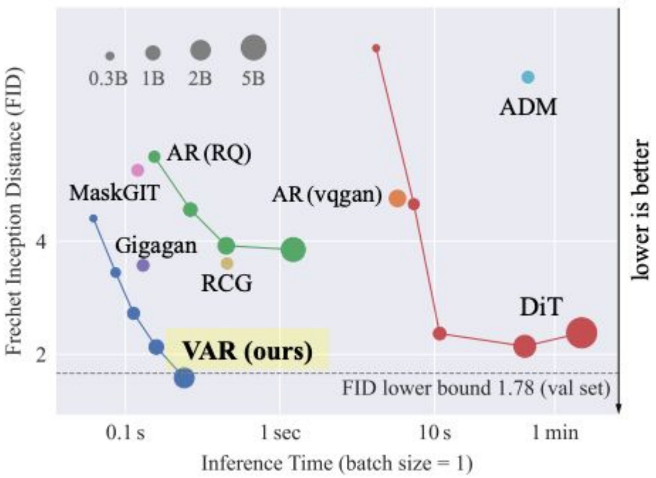
Contributions

- 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.
- An empirical validation of VAR models' Scaling Laws and zero-shot generalization potential, which initially emulates the appealing properties of large language models (LLMs).
- A breakthrough in visual autoregressive model performance, making GPT-style autoregressive methods surpass strong diffusion models in image synthesis for the first time.
- A comprehensive open-source code suite, including both VQ tokenizer and autoregressive model training pipelines, to help propel the advancement of visual autoregressive learning.

Introduction

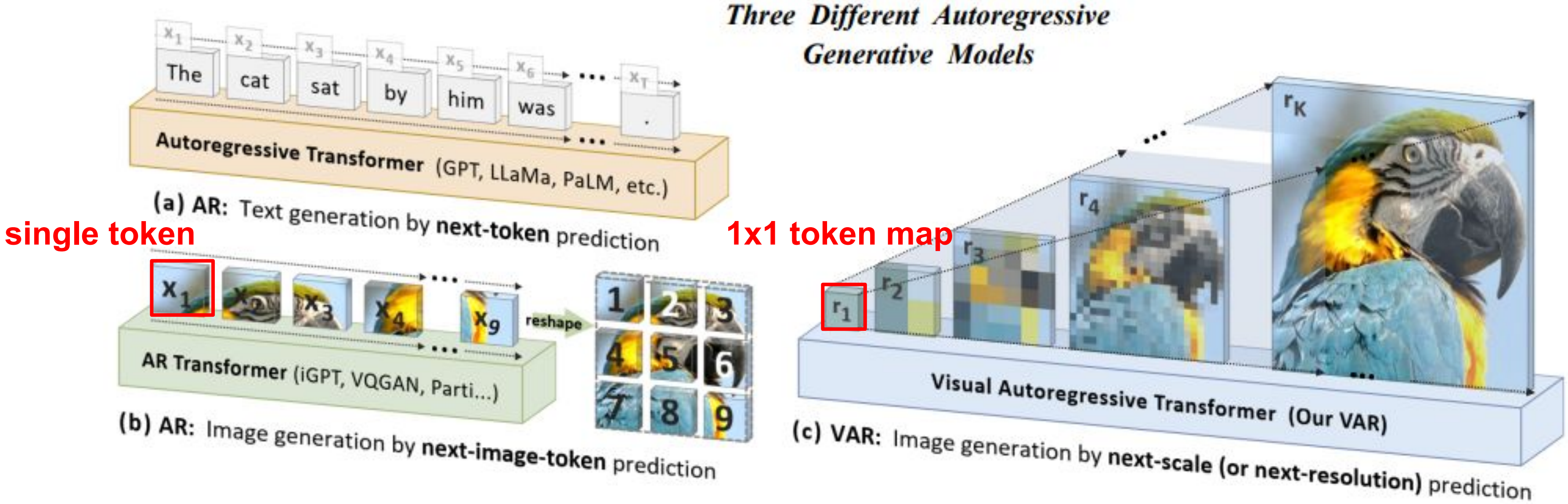
- **Problems**
  - scaling laws of the previous AR models remain underexplored.
  - the performance lags behind diffusion models.
  - the power of AR models in CV appears to be somewhat locked.





Introduction

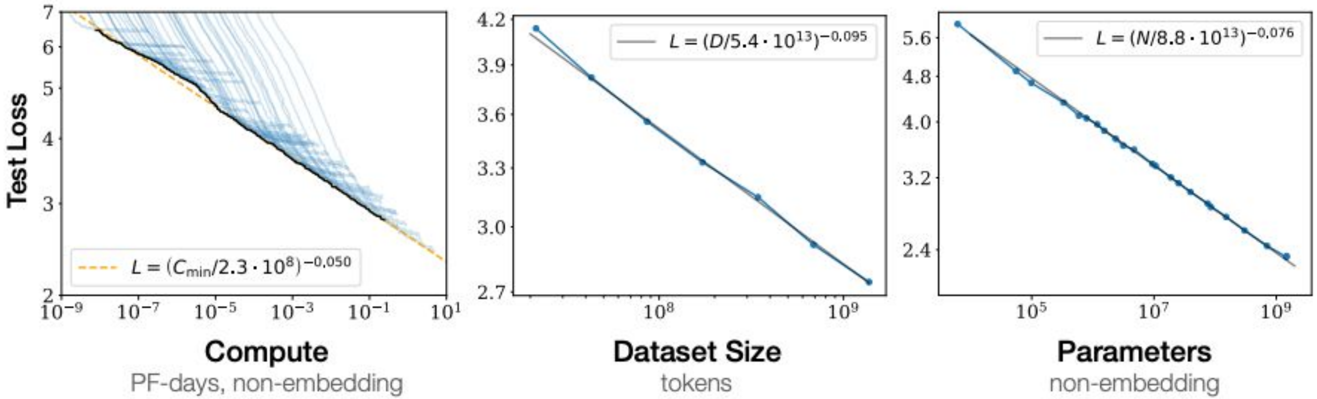
- **Proposed method**
  - reconsider how to order an image: human perception in hierarchical manner.
  - global -> local structure = multi-scale -> coarse-to-fine
  - next-token prediction -> next-scale prediction



-> Visual Autoregressive modeling (VAR):
predict the next higher resolution token map conditioned on all previous ones

Related works

- **Properties of large autoregressive language models**
  - scaling laws: the performance of LLMs x the growth of model, data, and computation -> inspired vision methods for generation

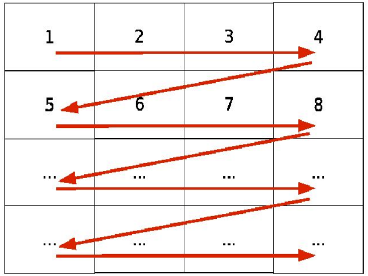


  - zero-shot generalization: perform tasks that it has not been trained on. In CV, there is a burgeoning interest in the zero-shot and in-context learning e.g.,) CLIP, Dinov2, LVM.

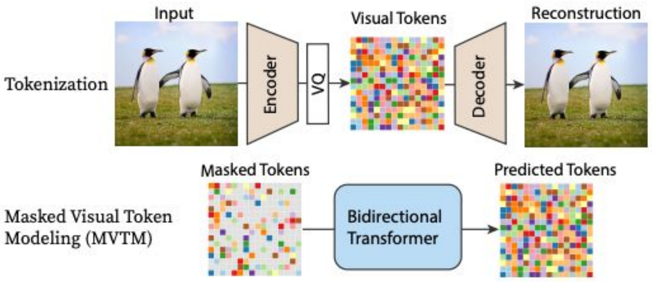
Fig. ref.: Scaling Laws for Neural Language Models, OpenAI

Related works

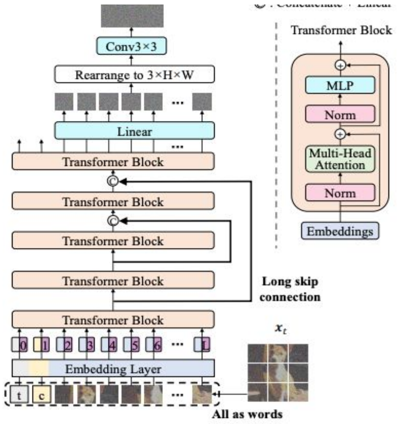
- **Visual generation**



Raster-scan



MaskGIT



U-ViT

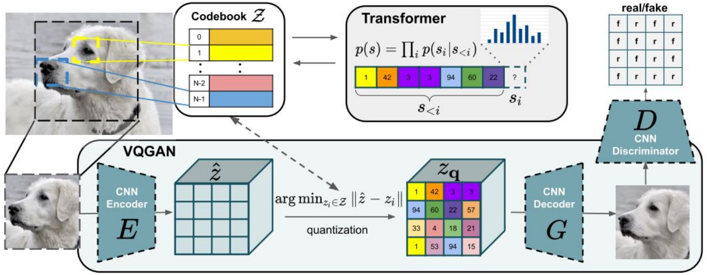
  - Raster-scan AR models: GPT-style. the encoding of 2D images into 1D token sequences. e.g.,) VQGAN, VQVAE-2, RQ-Transformer
  - Masked-prediction model: BERT-style. e.g.,) MaskGIT, MagViT-2, MUSE
  - Diffusion models: e.g.,) DiT, U-ViT

Method

- **AR modeling via next-token prediction**
  - Formulation: unidirectional token dependency assumption

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

  - Tokenization: 1) quantization 2) 1D ordering for unidirectional modeling. Once flattened, the autoencoder is fully trained, it will be used to tokenize images for subsequent training of AR model.



VQGAN - quantization

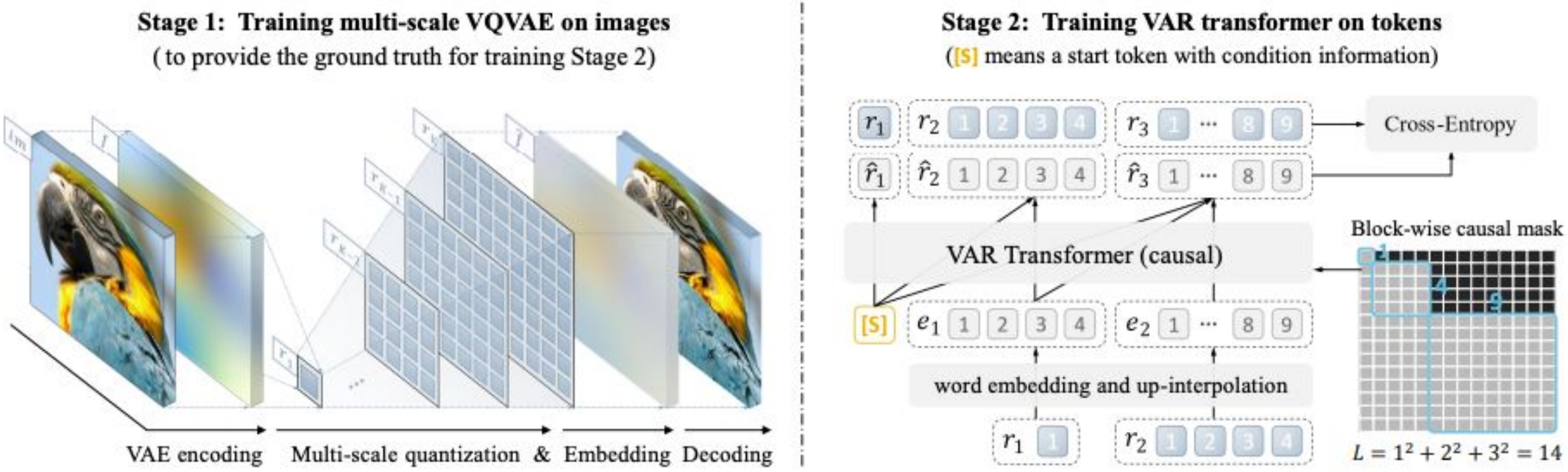
$$q^{(i,j)} = \left(\arg\min_{v \in [V]} \|\mathrm{lookup}(Z, v) - f^{(i,j)}\|_2 \right) \in [V],$$

Method

- **The weakness of vanilla autoregressive model**
  1) Mathematical premise violation
     a) after quantization and flattening, token sequence (x1, x2, . . . , xh×w) retains bidirectional correlations. <-> unidirectional assumption of AR
  2) Inability to perform some zero-shot generalization
     a) unidirectional modeling restricts generalizability in tasks requiring bidirectional reasoning.
  3) Structural degradation
     a) flattening disrupts the spatial locality in image feature maps
  4) Inefficiency
     a) generating an image tokens incurs O(n^2) autoregressive steps and computation cost O(n^6)

Method

- **VAR modeling via next-scale prediction**
  - Reformulation: next token prediction ->next scale prediction



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

r_k = token map at scale k containing h_k x w_k tokens.

Method

Tokenization

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

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 $\hat{im}$;

본 논문에서는 VAR을 학습시키기위한 이미지를 K multi-scale discrete token maps $R = (r_1, r_2, \cdots, r_K)$로 encode하는 새로운 multi scale quantization autoencoder를 개발한다. Encoding과 decoding 절차는 아래의 방식으로 이루어진다. 그리고 z_k를 $h_K \times w_K$로 upscaling을 할 때 정보 손실을 다루기위해, 본 논문에서는 K개의 extra convolution layers $\{\phi_k\}_{k=1}^{K}$를 사용한다.

ref.: Link

Method

- **VAR modeling via next-scale prediction**
  - Reformulation: next token prediction -> next scale prediction

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

- **Discussion**
  1) the mathematical premise is satisfied
     a) the process of getting r_k is solely related to r_<=k.
  2) the spatial locality is preserved
     a) no flattening operation and fully connected tokens in each r_k
  3) the complexity is reduced to O(n^4)
     a) parallel token generation in each r_k

Implementation details

VAR tokenizer

VAR transformer

Results

Type	Model	FID↓	IS↑	Pre↑	Rec↑	#Para	#Step	Time
GAN	BigGAN [13]	6.95	224.5	**0.89**	0.38	112M	1	–
GAN	GigaGAN [42]	3.45	225.5	0.84	**0.61**	569M	1	–
GAN	StyleGan-XL [74]	2.30	265.1	0.78	0.53	166M	1	0.3 [74]
Diff.	ADM [26]	10.94	101.0	0.69	0.63	554M	250	168 [74]
Diff.	CDM [36]	4.88	158.7	–	–	–	8100	–
Diff.	LDM-4-G [70]	3.60	247.7	–	–	400M	250	–
Diff.	DiT-L/2 [63]	5.02	167.2	0.75	0.57	458M	250	31
Diff.	DiT-XL/2 [63]	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.) [51]	3.49	215.5	–	–	502M	20	1.9 [51]
AR	VQVAE-2† [68]	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. [50]	7.55	134.0	–	–	3.8B	68	21
AR	RQTran.-re [50]	3.80	323.7	–	–	3.8B	68	21
VAR	VAR-*d*16	3.30	274.4	0.84	0.51	310M	10	0.4
VAR	VAR-*d*20	2.57	302.6	0.83	0.56	600M	10	0.5
VAR	VAR-*d*24	2.09	312.9	0.82	0.59	1.0B	10	0.6
VAR	VAR-*d*30	1.92	323.1	0.82	0.59	2.0B	10	1
VAR	VAR-*d*30-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**
  - Best FID/IS with remarkable speed
- **Compared with popular diffusion transformer**
  - outperform Diff transformers
- **Efficiency comparison**
  - VAR is around 20 times faster than VQGAN and ViT-VQGAN with more model params

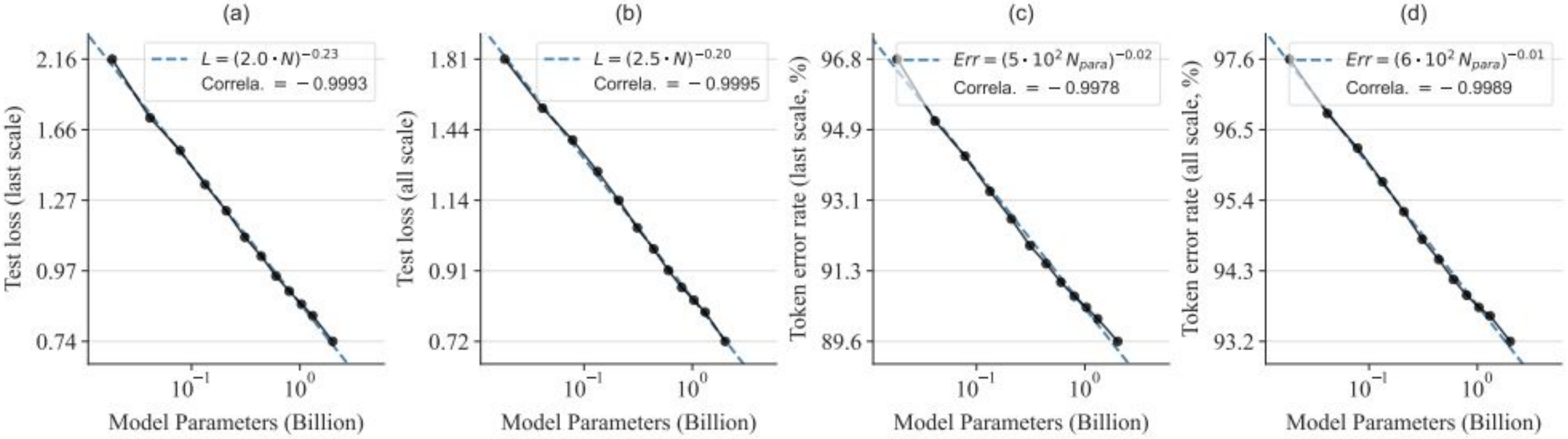
Results

- **Power-law scaling laws**
  - Background
    - Loss ∝ parameter counts N, training tokens T, and optimal training compute Cmin
  - Setup scaling VAR models
    - Train models across 12 different sizes, from 18M to 2B parameters on the ImageNet
    - For models of diffrent sizes, training with a maximum # of tokens 305 billions
    - Given sufficient token count T, focus on the scaling laws with model parameters N and optimal training compute Cmin

Results

- **Power-law scaling laws**
  - Test cross-entropy loss L and token prediction error rates Err
  - L and Err at the last next-scale autoregressive step (Fig (a) and (c)) and the global average (Fig (b) and (d))
  - The power-law scaling laws:

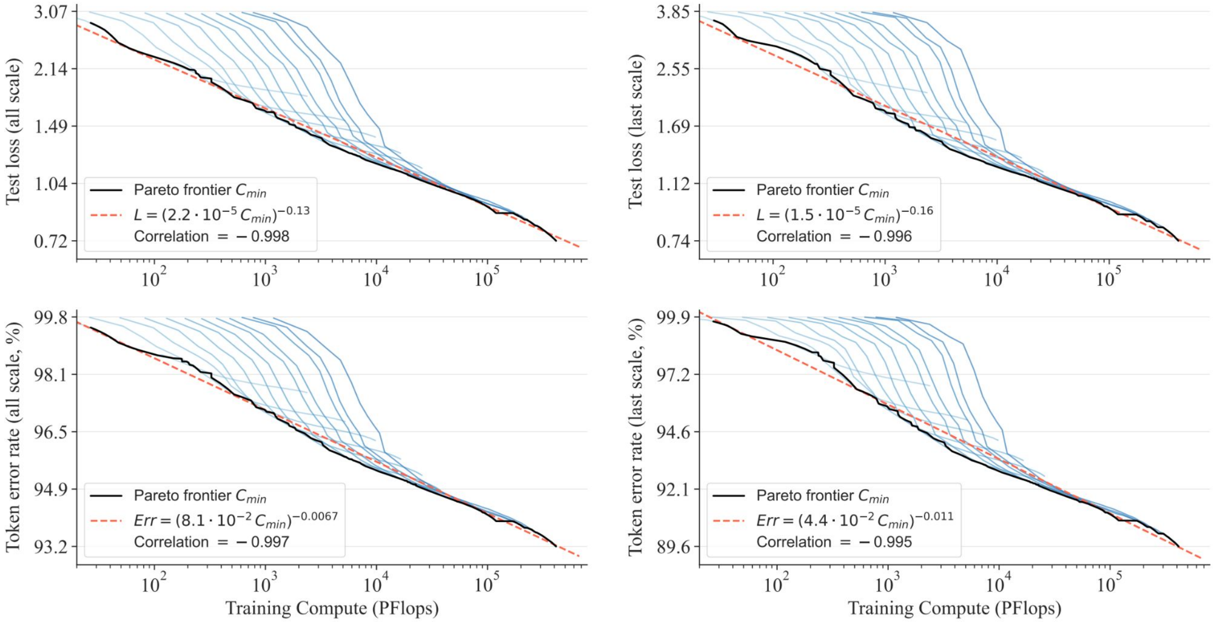
$$L_{\text{last}} = (2.0 \cdot N)^{-0.23} \quad \text{and} \quad L_{\text{avg}} = (2.5 \cdot N)^{-0.20} \qquad Err_{\text{last}} = (4.9 \cdot 10^2 N)^{-0.016} \quad \text{and} \quad Err_{\text{avg}} = (6.5 \cdot 10^2 N)^{-0.010}$$



→ Scaling up VAR transformers can continuously improve the model’s test performance

Results

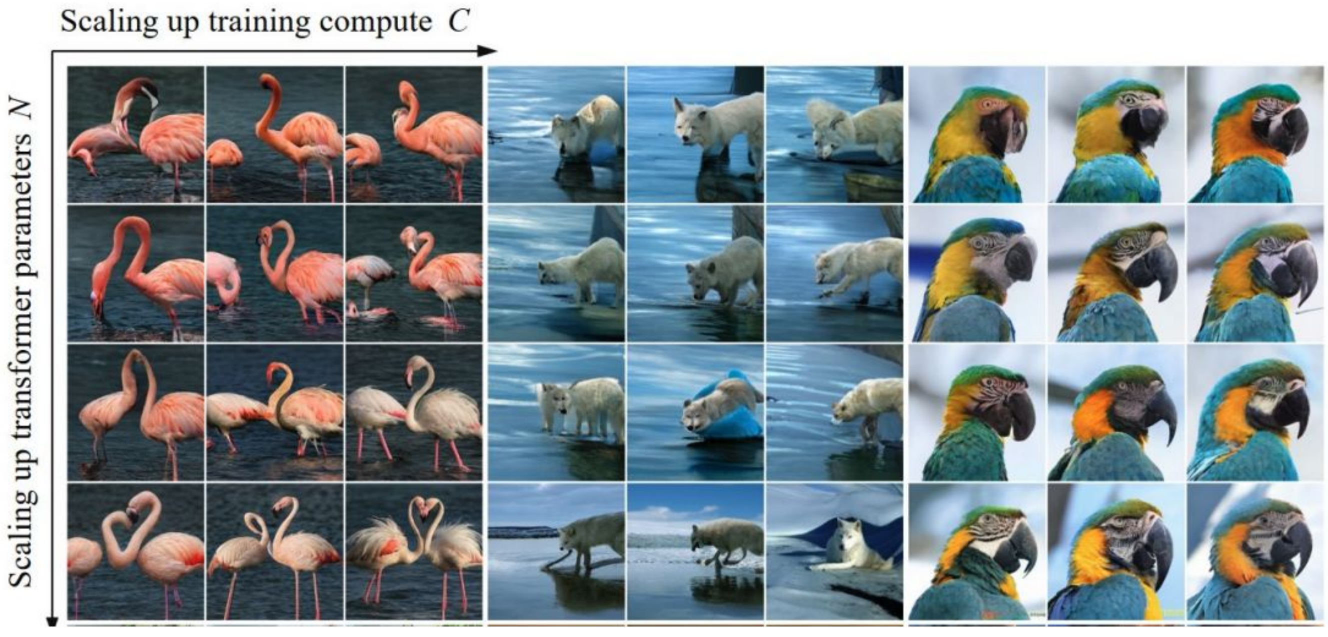
- **Power-law scaling laws**
  - Focus on scaling behavior of VAR transformers when increasing training compute C.



→ Trained with sufficient data, larger VAR transformers are more compute efficient because they can reach the same level of performance with less computation

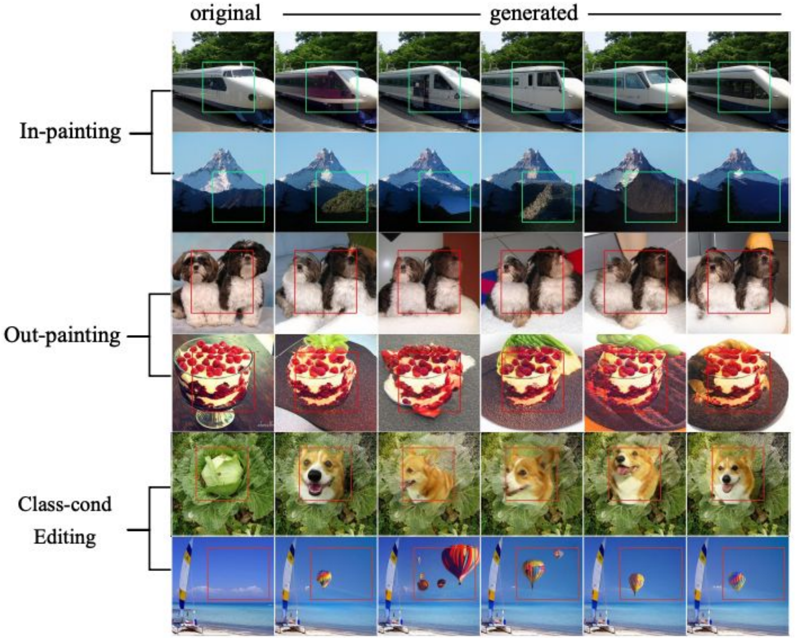
Results

- 4 different sizes (depth 6, 16, 26, 30) and 3 different training stages (20%, 60%, 100% of total training tokens)
- According to the scaling laws, larger transformers are able to learn more complex and fine-grained image distributions



Results

- **Zero-shot task generalization**
  - VAR has achieved decent results on these downstream tasks, substantiating the generalization ability of VAR
  - VAR model can produce plausible content that fuses well into the surrounding contexts, again



Ablation study

	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	✓	600	✗	0.016	4.64	-14.01
5	+CFG	310M	VAR-d16	✓	600	2.0	0.022	3.60	-15.05
5	+Attn. Norm.	310M	VAR-d16	✓	600	2.0	0.022	3.30	-15.35
6	+Scale up	2.0B	VAR-d30	✓	600	2.0	0.052	1.73	-16.85

Discussion

- Limitation
- 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, 59, 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.

→ VAR model depends on the performance of VQVQA tokenzer

- Future work
  1) Text prompt generation
  2) Video generation