MVAR:Visual Autoregressive Modeling with Scale and Spatial Markovian Conditioning

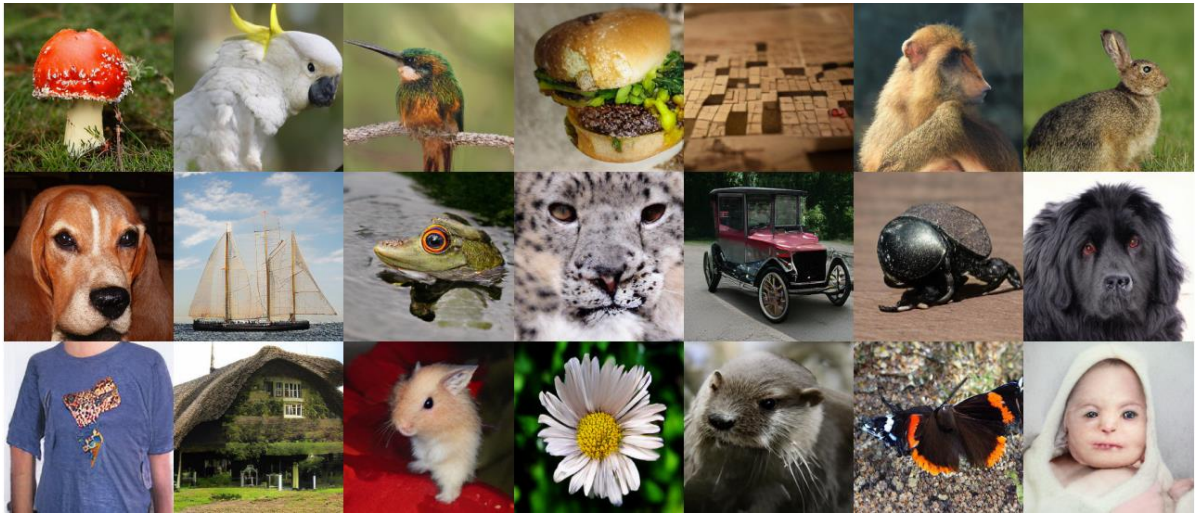
Jinhua Zhang, Wei Long, Minghao Han, Weiyi You, Shuhang Gu*

University of Electronic Science and Technology of China (UESTC)

CONTENTS

1.Background: Introduction of Image Generation

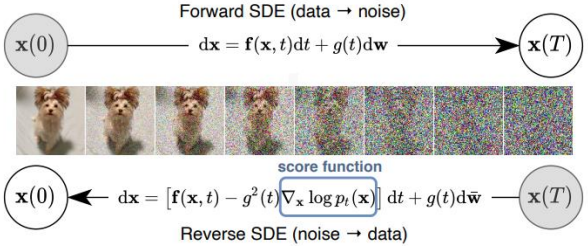
Image generation aims to learn the data distribution of images and generate new, realistic samples using generative models.



How to achieve image generation?

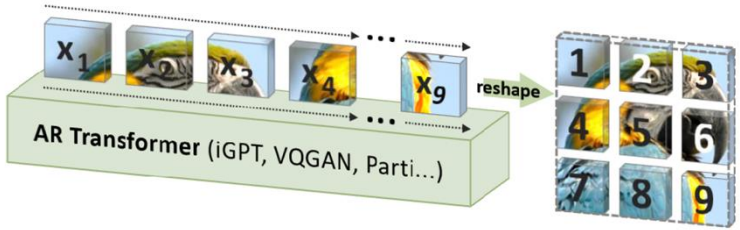
1.Background: Diffusion Model vs. Autoregressive Model

➢ **Denoising Diffusion Probabilistic Model [1]**



- ✓ Iteratively denoise random noise
- ✓ High image quality and diversity
- ✓ Slow sampling due to many steps

➢ **Autoregressive Image Generation Model [2]**



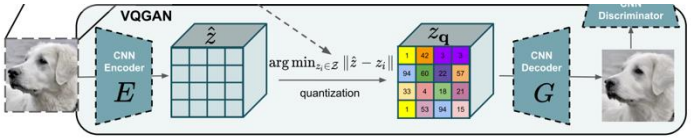
- ✓ Generate images by sequential token prediction
- ✓ Likelihood-based and stable training
- ✓ Long generation time for high-resolution images

Both paradigms suffer from **inefficient generation**, especially for high-resolution images.

[1] Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models[J]. Advances in neural information processing systems, 2020, 33: 6840-6851.
[2] Esser P, Rombach R, Ommer B. Taming transformers for high-resolution image synthesis[C] Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 12873-12883.

1.Background: AR Model (Tokenizer+Generative Model)

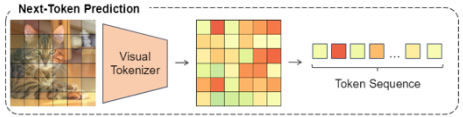
➢ Tokenizer (Encoder +Quantizer + Decoder)[2]



$$\hat{x} = G(z_{\mathbf{q}}) = G\left(\mathbf{q}(E(x))\right).$$

Higher compression ratio and utilization rate, and better reconstruction quality.

$$\mathcal{L}_{\mathrm{VQ}}(E, G, \mathcal{Z}) = \|x - \hat{x}\|^2 + \|\mathrm{sg}[E(x)] - z_{\mathbf{q}}\|_2^2 + \|\mathrm{sg}[z_{\mathbf{q}}] - E(x)\|_2^2.$$

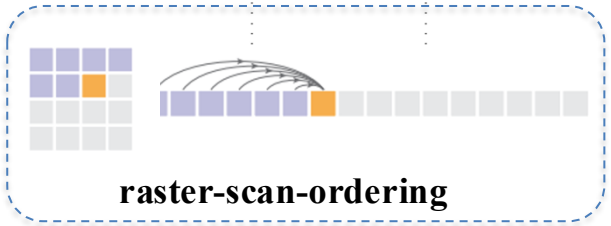


$$z_{\mathbf{q}} = \mathbf{q}(\hat{z}) := \left(\underset{z_k \in \mathcal{Z}}{\arg\min} \|\hat{z}_{ij} - z_k\|\right) \in \mathbb{R}^{h \times w \times n_z}.$$

Quantization: Find the most similar token index from the codebook.

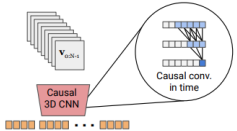
➢ Generative Model (Casual & No Casual)[3,4]

$$p(x) = \prod_{i=1}^{N} p(x_i | x_1, x_2, \ldots, x_{i-1}; \theta)$$

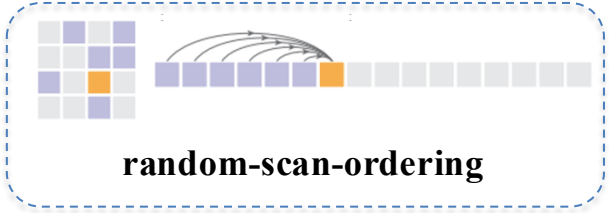


raster-scan-ordering

sequential token prediction.

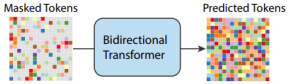


Casual Conv.



random-scan-ordering

scheduled parallel decoding.

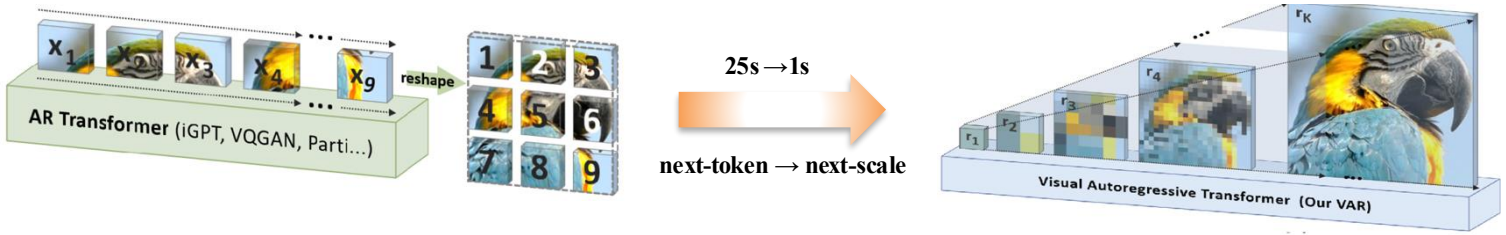


Bidirectional Attn.

[2] Esser P, Rombach R, Ommer B. Taming transformers for high-resolution image synthesis[C] Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 12873-12883.
[3] Chang H, Zhang H, Jiang L, et al. Maskgit: Masked generative image transformer[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 11315-11325.
[4] Yu L, Lezama J, Gundavarapu N B, et al. Language Model Beats Diffusion--Tokenizer is Key to Visual Generation[J]. arXiv preprint arXiv:2310.05737, 2023.

1.Background: Next-Token vs. Next-Scale Prediction

VAR[5] generates images by autoregressively predicting token maps from coarse to fine scales, enabling parallel generation within each scale.



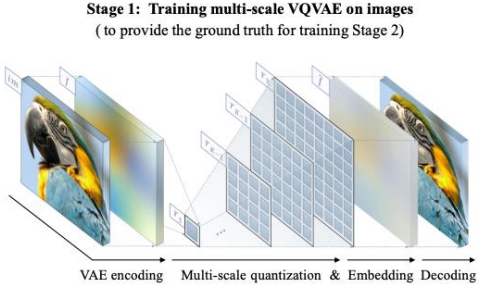
over scales rather than individual tokens.

- ✓ Multi-scale token maps better preserves the intrinsic 2D structure of images
- ✓ Autoregressive across scales and parallel generation within each scale speed the process of generation

[5] Tian K, Jiang Y, Yuan Z, et al. Visual autoregressive modeling: Scalable image generation via next-scale prediction[J]. Advances in neural information processing systems, 2024, 37: 84839-84865.

1.Background: Visual AutoRegressive (VAR)

➢ Multi-Scale Residual Quantization



multi-scale VQ autoencoder

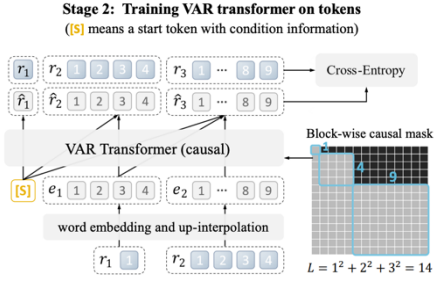
Algorithm 1: Multi-scale VQVAE Encoding

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 $\hat{im}$;

➢ Multi-Scale Parallel Training



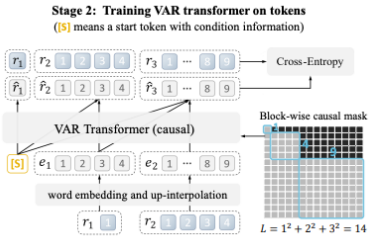
$$p(r_1, r_2, \ldots, r_K) = \prod_{k=1}^{K} p(r_k \mid r_1, r_2, \ldots, r_{k-1}),$$

```
for si in range(SN - 1):
    if si >= self.prog_ed:
        break  # diffusion-like training: supported output from 0-prog_ed
    h_BCHW = F.interpolate(
        self.embedding(gt_ms_idx_Bl[si])
        .transpose_(1, 2)
        .view(B, C, pn_next, pn_next),
        size=(H, W),
        mode="bicubic",
    )
    f_hat.add_(self.quant_resi[si / (SN - 1)](h_BCHW))
    pn_next = self.v_patch_nums[si + 1]
    next_scales.append(
        F.interpolate(f_hat, size=(pn_next, pn_next), mode="area")
        .view(B, C, -1)
        .transpose(1, 2)
    )
```

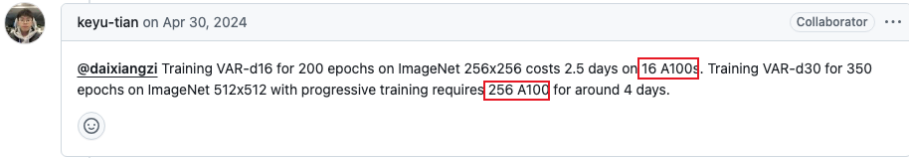
[5] Tian K, Jiang Y, Yuan Z, et al. Visual autoregressive modeling: Scalable image generation via next-scale prediction[J]. Advances in neural information processing systems, 2024, 37: 84839-84865.

1.Background: Drawback of VAR's Scale Dependency

Although VAR is capable of rapid image generation, it incurs substantial training costs.



- **Inference Time: 25s →1s**



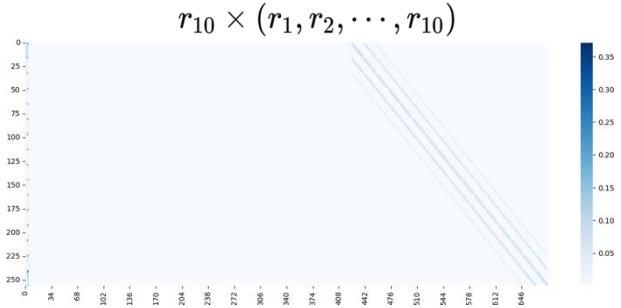
- **Token Length of Training:256 → 680**

- **Redundant scale dependence (token length from 256 to 680)**
- **High GPU memory usage (16×A100 80G)**

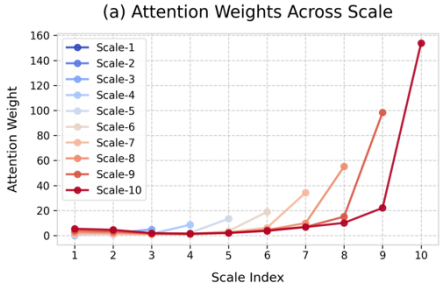
[5] Tian K, Jiang Y, Yuan Z, et al. Visual autoregressive modeling: Scalable image generation via next-scale prediction[J]. Advances in neural information processing systems, 2024, 37: 84839-84865.

2. Motivation: Redundancy as the Bottleneck of VAR

Why VAR requires larger GPU memory: Attention redundancy across Scales.



(a) Inter-scale attention map



(b) Overall attention weight distribution across scales

Qualitative and quantitative results analysis of VAR attention patterns across **scales**.

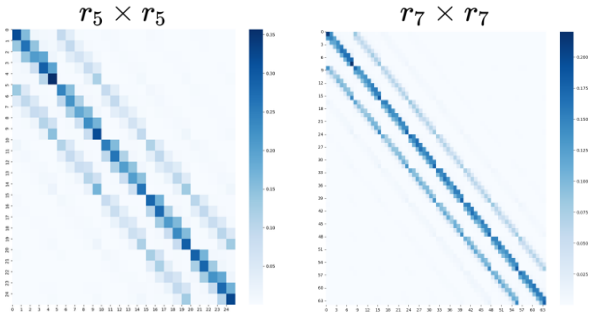
- Observation 1: Attention weights in the VAR model exhibit **scale redundancy**, indicating that each scale mainly depends on its **adjacent scales**.

Implication: Attending to all scales is unnecessary and memory-inefficient.

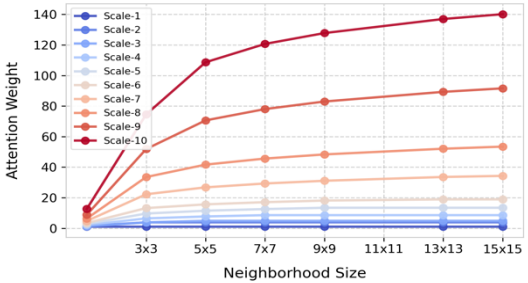
[6] Zhang J, Long W, Han M, et al. MVAR: Visual Autoregressive Modeling with Scale and Spatial Markovian Conditioning[J]. arXiv preprint arXiv:2505.12742, 2025.

2. Motivation: Redundancy as the Bottleneck of VAR

Why VAR requires larger GPU memory: Attention redundancy across Spatial.



(a) Adjacent-scale attention map



(b) Attention weights across varying neighborhood sizes

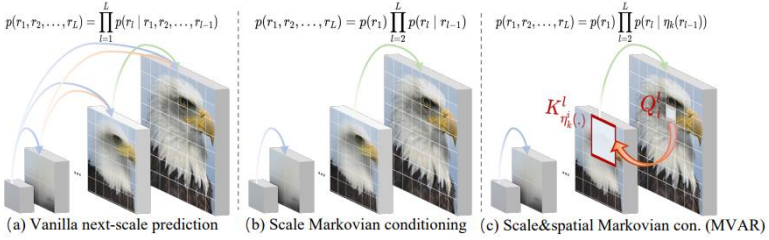
Qualitative and quantitative results analysis of VAR attention patterns across **spatial**.

- Observation 2: Attention weights in the VAR model exhibit **spatial redundancy**, with dependencies concentrated within **local spatial neighborhoods**.

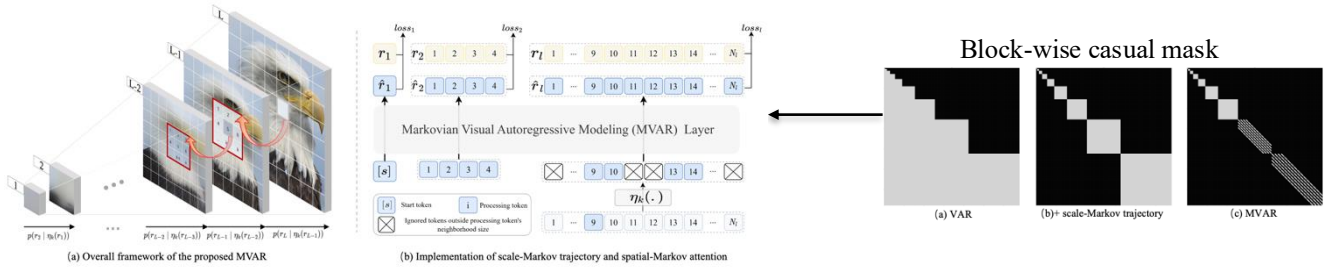
Implication: Global attention introduces redundant spatial computation.

[6] Zhang J, Long W, Han M, et al. MVAR: Visual Autoregressive Modeling with Scale and Spatial Markovian Conditioning[J]. arXiv preprint arXiv:2505.12742, 2025.

3.Method : Scale and Spatial Markovian Conditioning



Introducing **scale and spatial Markov assumption** to multi-scale autoregressive image generation[6]



- Enabling the adoption of a parallel training strategy using only **eight NVIDIA RTX 4090** GPUs.
- Reducing the computational complexity of attention calculation from $\mathcal{O}(N^2)$ **to** $\mathcal{O}(Nk)$.

[6] Zhang J, Long W, Han M, et al. MVAR: Visual Autoregressive Modeling with Scale and Spatial Markovian Conditioning[J]. arXiv preprint arXiv:2505.12742, 2025.

3.Results: Better Performance, Lower GPU Memory

- **Lower GPU memory footprint and faster training/inference speed**

Methods	Time (s)↓	GFLOPs↓	KV Cache↓	Memory↓	Train Speed↓	Train Memory↓	FID↓	IS↑	Precision↑	Recall↑
VAR-*d16*	0.34	43.61	5440M	10882M	0.99	34319M	3.55	280.4	0.84	0.51
MVAR-*d16*†	0.27	35.44	**0**	3846M (**2.8**×)	0.61 (**1.6**×)	20676M	3.40	297.2	0.86	0.48
VAR-*d20*	0.52	81.52	8500M	16244M	1.35	48173M	2.95	302.6	0.83	0.56
MVAR-*d20*†	0.45	68.75	**0**	5432M (**3.0**×)	0.79 (**1.7**×)	27665M	2.87	295.3	0.86	0.52
VAR-*d24*	0.81	136.63	12240M	23056M	–	OOM	2.33	312.9	0.82	0.59
MVAR-*d24*†	0.71	118.25	**0**	7216M (**3.2**×)	0.91	38579M	2.23	300.1	0.85	0.56

Table : Quantitative comparison between VAR and our MVAR.

- **Superior image quality**

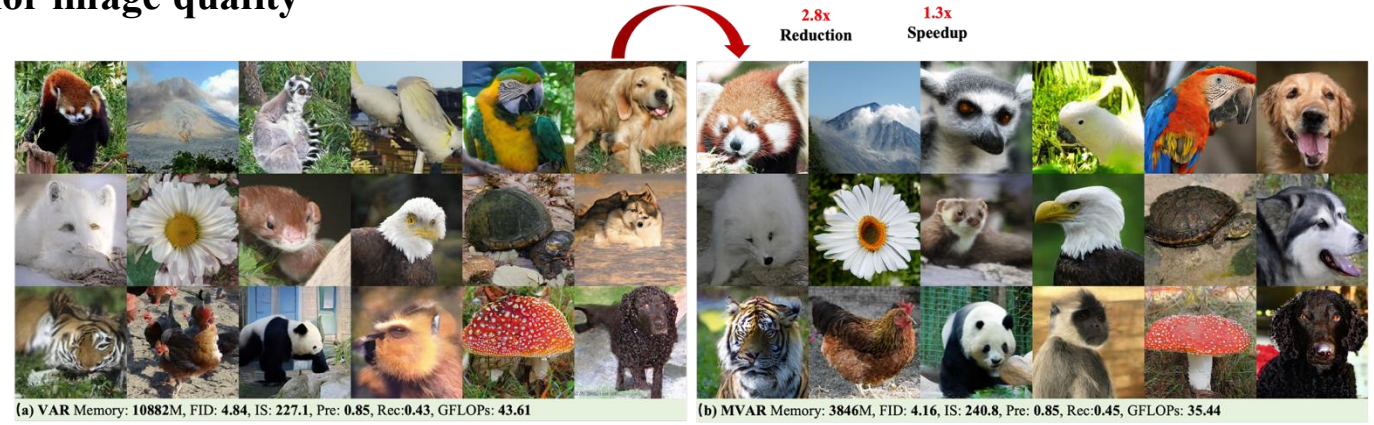


Figure : Qualitative results of MVAR.

DIG

Thank You!

University of Electronic Science and Technology of China (UESTC)

Jinhua Zhang (张进华)

https://nuanbaobao.github.io/