MVAR: Visual Autoregressive Modeling with Scale and Spatial Markovian Conditioning

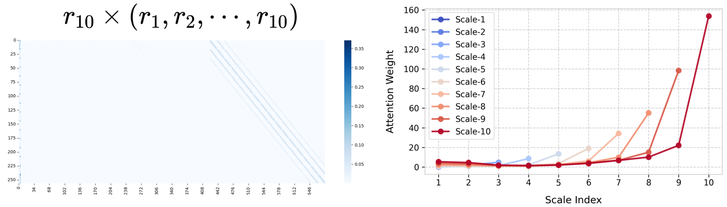
Jinhua Zhang, Wei Long, Minghao Han, Weiyi You, Shuhang Gu*
University of Electronic Science and Technology of China

ICLR

Markovian Visual AutoRegressive

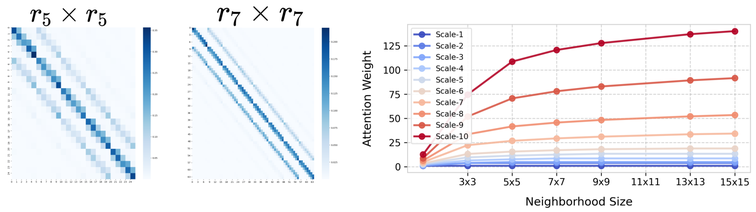
- VAR condition each scale on all previous scales and require each token to consider all preceding tokens, exhibiting scale and spatial redundancy.
- We introduces **scale and spatial Markov assumptions** to reduce the complexity of conditional probability modeling .
- We reduce the computational complexity of attention calculation from $\mathcal{O}(N^2)$ to $\mathcal{O}(Nk)$, enabling training with just eight NVIDIA RTX 4090 GPUs and eliminating the need for KV cache during inference.

Redundancy as the Bottleneck of VAR



(a) Inter-scale attention map (b) Overall attention weight distribution across scales

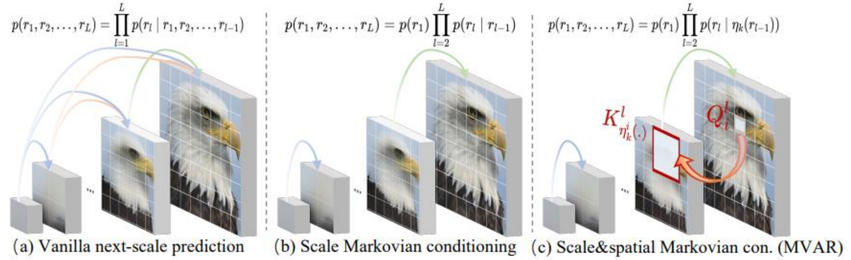
➢ *Observation 1*: Attention weights in the VAR model exhibit **scale redundancy**, indicating that each scale mainly depends on its **adjacent scales**.



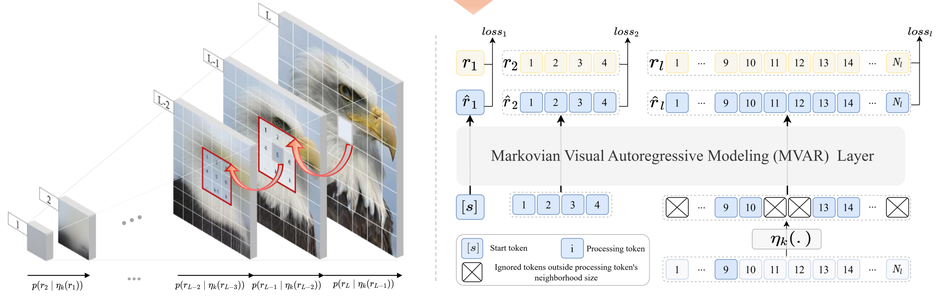
(a) Adjacent-scale attention map (b) Attention weights across varying neighborhood sizes

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

Introducing Scale and Spatial Markov Assumption to VAR



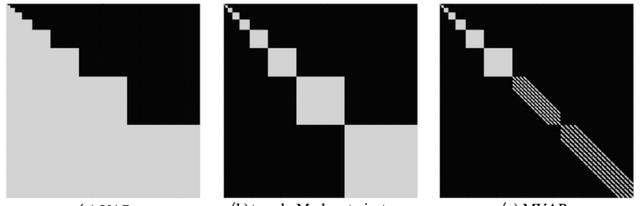
(a) Vanilla next-scale prediction (b) Scale Markovian conditioning (c) Scale&spatial Markovian con. (MVAR)



(a) Overall framework of the proposed MVAR (b) Implementation of scale-Markov trajectory and spatial-Markov attention

First, a **scale-Markov trajectory** predicts r_l using only its adjacent scale r_{l-1}, discarding all earlier scales. This allows for parallel training across scales using a standard cross-entropy loss loss . Second, a **spatial-Markov attention** mechanism restricts attention to a local neighborhood of size k, reducing computational complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(Nk)$.

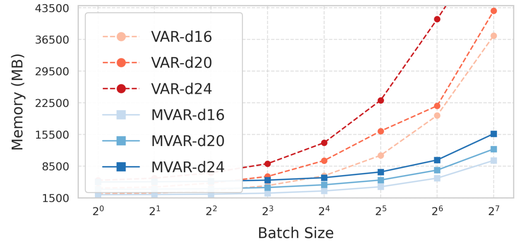
Block-wise Casual Mask



(a) VAR (b)+ scale-Markov trajectory (c) MVAR

VAR employs a full causal mask to model $p(r_l \mid r_{<l})$. In MVAR, scales r_1 to r_8 use a diagonal-pattern mask to model $p(r_l \mid r_{l-1})$, scales r_9 and r_{10} are generated using custom CUDA kernels to model $p(r_l \mid \eta_k(r_{l-1}))$.

Lower GPU Memory Footprint During Inference



MVAR reduces memory consumption by **4.2×** compared to VAR-d24 (9,860MB vs. 40,971MB at batch size 64).

Superior Image Quality

(a) VAR Memory: **10882**M, FID: **4.84**, IS: **227.1**, GFLOPs: **43.61**

(b) MVAR Memory: **3846**M, FID: **4.16**, IS: **240.8**, GFLOPs: **35.44**

2.8x Reduction
1.3x Speedup

Code and Models

Code Paper Lab