MoSs: Mixture of Scales for Efficient High-Resolution Autoregressive Image Generation

Yaoxiu Lian[1], Hao Liang[1*], Zhihong Gou[1], Yijia Zhang[1], Jiaming Xu[1], Guohao Dai[1, 2, 3*], Ningyi Xu[1]

[1]Shanghai Jiao Tong University
[2]Infinigence-AI
[3]Shanghai Innovation Institute (SII)

Abstract

Since next-scale prediction was introduced as a new paradigm for autoregressive image generation, it has attracted extensive research interest. By progressively increasing resolution in a draft-to-refinement process, next-scale prediction demonstrates great potential in both generation quality and efficiency. However, at high resolutions, this paradigm faces a fundamental challenge: token sequences grow quadratically and accumulate across multiple scales, resulting in a critical performance bottleneck. Our systematic study reveals two key observations: (1) most image regions stabilize during early drafting stages, rendering subsequent full-scale refinement token-inefficient; and (2) different scales inherently present efficiency-fidelity trade-offs, suggesting that adaptive token dispatch across scales can concentrate computational resources where they yield the greatest quality gains. Motivated by these insights, we propose a training-free **M**ixture **o**f **S**cale**s** (**MoSs**) method for efficient high-resolution autoregressive image generation. MoSs breaks the strict causal dependency across scales in the final refinement steps by parallelizing scales of different resolutions, with each scale responsible for a subset of spatial regions. A lightweight frequency-based token dispatcher analyzes the drafted image and assigns regions to the appropriate scale. The outputs are then composited over the draft to produce the final high-resolution image. The scale-mixture method achieves substantial efficiency improvements with minimal impact on generation quality across various models. For instance, our implementation achieves **2.05-4.96× speedup** on transformer backbone, up to **85.62% KV cache reduction**, while incurring only **0.1-2.4%** loss on GenEval quality metrics, as demonstrated on the state-of-the-art Infinity model.

Introduction

Text-to-image generation has emerged as a central challenge in multimodal intelligence, with wide-ranging applications in content creation, augmented reality, human-computer interaction, and digital media. While diffusion-based [Ho, Jain, and Abbeel 2020, Rombach et al. 2022, Chen et al. 2024a] approaches currently dominate due to their superior perceptual quality, autoregressive (AR) models [Wang et al. 2024, Esser, Rombach, and Ommer 2021, Tian et al. 2024] have recently shown competitive potential, particularly with the emergence of large-scale transformers. In particular, the paradigm of next-scale (or next-resolution) prediction, also known as Visual Autoregressive Modeling (VAR), has achieved remarkable success.

Unlike traditional raster-scan autoregression that generates tokens sequentially across spatial positions, VAR reformulates image synthesis as a coarse-to-fine process where complete token maps are predicted at progressively higher resolutions across different "scales". This formulation not only enhances generalization and image quality but also enables parallel token prediction within each scale, significantly improving computational efficiency. This paradigm has enabled AR transformers to surpass diffusion models in both generation quality and controllability across multiple benchmarks, subsequently catalyzing extensive research efforts to further explore its potential [Tang et al. 2024, Han et al. 2024, Gao et al. 2025, Qu et al. 2025, Zhuang et al. 2025].

By viewing VAR as an early drafter and a later refiner in Fig. 1, we find the drafter extremely fast processing low-resolution token matrices, but the refiner suffers in performance at-high resolution scales. As resolution increases, the number of tokens per scale grows quadratically over the spatial dimension n. In a typical next-scale prediction paradigm, such as Infinity [Han et al. 2024], generating a 1024×1024 image involves 13 successive resolution scales. Throughout this process, a total of 10,521 tokens are decoded and 6,425 tokens need to be cached. Notably, the last four scales alone decode 8,000 tokens and store 4,928 tokens in the KV cache, accounting for over **80%** of the inference latency and up to **77%** of total KV cache consumption, thus constituting the primary performance bottleneck.

To help mitigate the performance problem, we conduct a systematic analysis of next-scale prediction model behavior and identify two key insights: **First, the early stages generate a very decent foundation draft, and the refinement stages further provide detailed improvements.** We observe that initial "drafter" stages achieve decent perceptual quality with far fewer tokens, whereas subsequent "refiner" stages with higher computation provide only marginal quality improvement. **Second, refining itself is spatial-heterogeneous and highly correlated with frequency.**

*Corresponding author: lianghao@sjtu.edu.cn, daiguohao@sjtu.edu.cn

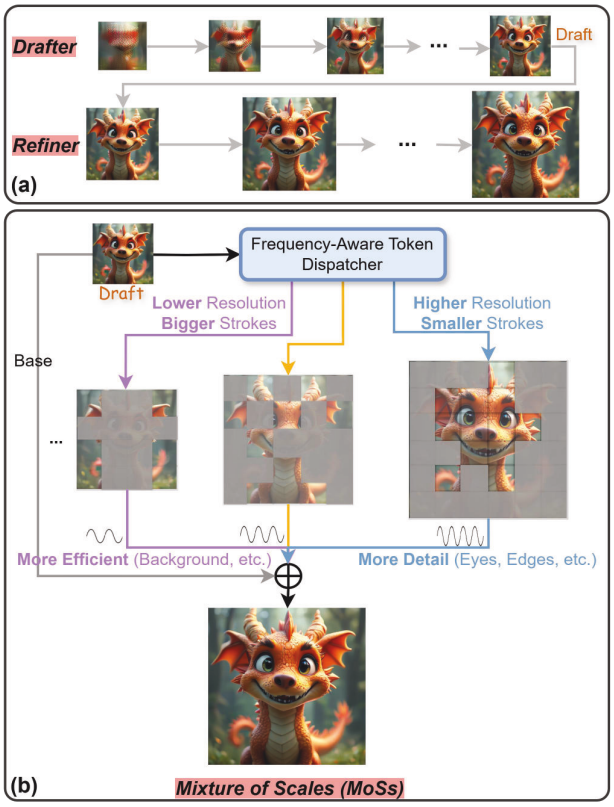


Figure 1: (a) standard next-scale prediction method of drafter-refiner. (b) mixture of scales method of drafter-MoSs.

Some image regions (background, bulks) have stabilized early in the drafter, while high-frequency regions (textures and edges) require fine-grained updates. The need for refinement varies significantly across spatial regions. Prior work in image encoding and decoding has shown that the frequency characteristics of image regions are strongly correlated with their computational requirements [Siddeq and Rodrigues 2017, Rhee et al. 2022].

These insights reveal substantial redundancy in the refinement process, indicating significant opportunities to exploit spatial sparsity for enhanced computational efficiency. We introduce a novel refinement framework, the **Mixture of Scales (MoSs)**, which is training-free and can be seamlessly integrated into various existing VAR models. In this approach, the drafting stages of the original VAR model remain unchanged to generate the initial coarse draft. Subsequently, MoSs employs a frequency-aware token dispatcher that partitions image regions based on their refinement requirements, as determined by frequency analysis. Specifically, regions with higher spatial frequency are assigned to finer resolution scales, while those with lower frequency are mapped to coarser scales.

These regions, represented as distinct token sets, constitute a mixture of resolution scales. Unlike the original VAR paradigm, which requires multiple sequential refinement stages, MoSs reuses the transformer backbone to process the mixed-resolution token sets in parallel, reducing computational overhead to a single forward pass. The output tokens from each scale are scattered to empty token maps corresponding to their assigned resolutions and then interpolated to the final output size. Finally, the parallelly refined results are composited onto the initial draft using the image decoder, producing the final high-resolution image.

Our contributions are summarized as follows:

1. We identify the key performance limitations of the next-scale prediction paradigm and provide comprehensive qualitative and quantitative analyses that motivate our Mixture of Scales (MoSs) approach.
2. We propose a training-free MoSs framework that adaptively distinguishes, partitions, and parallelizes the computation of spatial regions in VAR-based image generation, thereby achieving substantial improvements in both efficiency and image quality.
3. MoSs can be seamlessly integrated into VAR-like models, and our implementation achieves **2.05-4.96× speedup**, up to **85.62% KV cache reduction**, incurring only **0.1-2.4%** loss on GenEval, based on the state-of-the-art Infinity model.

Related Work

Autoregressive Image Generation Early autoregressive image generation models synthesize images by predicting pixels sequentially in raster scan order [Gregor et al. 2014, Van Den Oord, Kalchbrenner, and Kavukcuoglu 2016, Van den Oord et al. 2016, Parmar et al. 2018, Chen et al. 2018, 2020], using architectures such as RNNs, masked CNNs, or Transformers. Subsequent approaches [Van Den Oord, Vinyals et al. 2017, Razavi, Van den Oord, and Vinyals 2019, Ramesh et al. 2021, Esser, Rombach, and Ommer 2021] compress spatial image patches into tokenized latent spaces, typically arranged in raster order. More recent work [Chang et al. 2022, Li et al. 2023, 2024a, Fan et al. 2024] propose bidirectional attention mechanisms that generate masked token subsets iteratively, breaking the raster order constraint.

Most recently, VAR and subsequent work [Tian et al. 2024, Han et al. 2024] introduce scale-wise autoregression, progressively increasing image resolution and generating tokens at coarser-to-finer scales. This redefines generation order between resolution scales rather than pixels or patches. FratalAR [Li et al. 2025] approaches next-scale prediction as a fractal generation process with parallel synthesis and local attention. Next-scale prediction has expanded across text-to-image synthesis [Han et al. 2024, Ma et al. 2024, Tang et al. 2024, Zhang et al. 2024], conditional generation [Li et al. 2024b,c], and other modalities like audio and 3D [Qiu et al. 2024, Zhang, Xiong, and Xu 2024].

Autoregression Efficiency Autoregressive transformers for image generation follow scaling laws [Tian et al. 2024, Shukor et al. 2025] analogous to large language models [Henighan et al. 2020, Kaplan et al. 2020]. However, quadratic self-attention complexity [Vaswani et al. 2017] is

(a) Sequence Length vs Quality Gain

(b) Drafter stage + Refinement stage

Figure 2: (a) **Quaquatically accumulating sequence length versus diminishing quality gains**, indicating quality gain per token drops significantly during the refinement stages. (b) **Qualitative demonstration of drafter and refiner results**, the drafter is the first 8 stages of Infinity, and the refiner is the last 5 stages.

more pronounced for 2D image tokens than 1D text. Vision transformers typically partition images into patches [Dosovitskiy et al. 2020, Liu et al. 2021], while token-reduction strategies through pruning or merging further shorten sequences [Bolya et al. 2022, Rao et al. 2021, Liang et al. 2022]. Speculative decoding methods have been adapted for parallel token prediction [Jang et al. 2024, Teng et al. 2024].

VAR [Tian et al. 2024] significantly improves efficiency by reducing time complexity from n^6 to n^4, where n denotes the spatial dimension of the token map. To address the performance degradation observed in refinement scales of VAR, CoDe [Chen et al. 2024b] introduces a collaborative inference using both large and small models. LiteVAR [Xie et al. 2024] builds on VAR by identifying locality properties in visual attention patterns, drawing parallels to efficient mechanisms developed for large language models [Xiao et al. 2023]. Most recently, FastVAR [Guo et al. 2025] enhances decoding efficiency by pruning redundant tokens in the later stages, leading to additional performance gains. While both FastVAR and our approach leverage frequency analysis to determine which tokens to process, a key difference lies in our method's core innovation: we utilize spatial partitioning to enable parallel multi-scale acceleration.

Methodology

Preliminary

Visual Autoregressive Modeling (VAR) formulates image synthesis as a multi-stage, coarse-to-fine process, where each stage corresponds to a specific resolution scale. Given K scales, VAR models the joint distribution of multi-scale outputs $\{\mathbf{r}_1, ..., \mathbf{r}_K\}$ as follows:

$$p(\mathbf{r}_1, ..., \mathbf{r}_K) = \prod_{k=1}^{K} p(\mathbf{r}_k \mid \mathbf{r}_{<k}), \tag{1}$$

where $\mathbf{r}_k$ denotes the token map (or residual) at the k-th scale, and $\mathbf{r}_{<k} = \{\mathbf{r}_1, ..., \mathbf{r}_{k-1}\}$.

At each stage k, $\mathbf{r}_k$ has resolution $h_k \times w_k$, which increases with k. The cumulative output R_k at stage k is constructed by upsampling each residual to the final resolution and summing them. After generating R_k, the output is downsampled and fed as context to the next stage $k+1$. This autoregressive chain enforces strict sequential dependency across stages, that every scale refines the output from its previous stage. All the tokens in the same scale are processed in parallel, regardless of whether the token has actually stabilized.

Text prompts, encoded by language models, can be consumed before all stages to generate R_1 or used by cross-attention blocks in the transformer, depending on the specific model design. KV-Caching is usually performed for all tokens processed by all previous stages, and no masking is necessary at inference time. VAR is extremely efficient at the early drafting stage with a low-resolution token matrix, while late-stage high-resolution refinement is both computationally and memory-expensive.

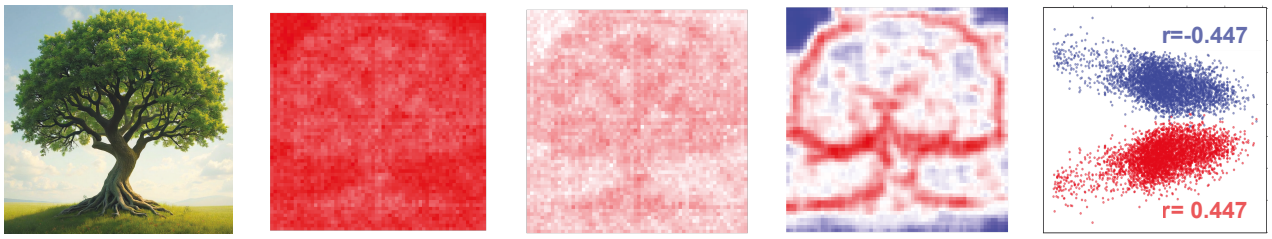
Key Observation

To elucidate the sources of inefficiency in late-stage refinement, we conduct a stage-wise empirical analysis over several pretrained VAR models. Our study reveals two fundamental findings that motivate the MoSs framework.

Observation 1: Imbalance between contribution and cost. As illustrated in Fig. 2(a), the GenEval score increases significantly during the initial (early) scales but quickly saturates at later stages, even as the number of tokens continues to grow quadratically. To quantify the contribution of each generation stage to the final image, we project each stage's token embeddings onto the final output token vector. For token t, the contribution from a given stage S_k is defined as:

$$C_t = \frac{\langle h_t^k, h_t^{final} \rangle}{|h_t^{final}|^2}, \tag{2}$$

where h_t^k and h_t^{final} denote the token embeddings at stage S_k and the final output respectively. Using this formulation, we construct token-wise contribution heatmaps. Our results show that the drafter stage (the first 8 stages) achieves an average contribution score of 0.615, whereas the refiner stage (the last 5 stages) achieves only 0.385. Despite this, the drafter accounts for merely 9% of total FLOPs, while the refiner consumes over 90%. Fig. 2(b) demonstrates that an 8-scale drafter already constructs the majority of the image's global layout, such as the structure and background,



(a) Generated Image (b) Drafter Contribution (c) Refiner Contribution (d) Frequency Analysis (e) Freq-Contribution Corr

Figure 3: A sample image generated by the Infinity model. (a) **Generated Image**, prompt: "Area of Effect Healing tree, realism, welcoming, marketable". (b) **Drafter Contribution**, per-token contribution of drafting stages over all stages. (c) **Refiner Contribution**, per-token contribution of refining stage over all stages. (d) **Frequency Analysis**, token-wise frequency analysis based on local region variance. (e) **Frequency-Contribution Analysis**, correlation between token frequency and drafter/refiner contribution, indicating drafter mostly work on low-frequency regions, while refiner mostly work on high-frequency regions.

whereas the later 5-scale refiner is primarily responsible for sharpening fine-grained details, such as textures and edges. This indicates that many regions of the image have already stabilized after the drafter, yet are repeatedly processed in the refinement stages, resulting in substantial computational redundancy.

Observation 2: Frequency-sensitive spatial heterogeneity. To examine how refinement demand varies across spatial regions, we analyze per-pixel contributions in the final few stages. As shown in Fig. 3(c), the contributions are unevenly distributed: lighter pixels require more refinement, while darker ones are sufficiently resolved. We further correlate the contribution values with local pixel frequencies and find a clear positive relationship ($r = 0.447$, Fig. 3(e)). Local pixel frequencies are calculated using Eq. 3 as introduced later. We find this decent correlation varies but is ubiquitous across many VAR models and prompts. Furthermore, Fig. 3(b) shows that low-frequency regions are primarily refined by the drafter, while high-frequency regions are refined by later stages. This complements Observation 1 and confirms a frequency-based division of labor in VAR. It also suggests that spatial regions can be refined independently based on their frequency characteristics: low-frequency regions with coarse details can be refined at lower resolution, whereas high-frequency regions with fine details require finer scales.

These two observations jointly motivate a frequency-aware region-to-stage assignment strategy: instead of enforcing full serial decoding, each pixel is processed exclusively by the refinement stage most appropriate to its frequency demand. This principle forms the foundation of our MoSs framework, which we will explain in detail in the following section.

Mixture-of-Scales (MoSs) Architecture

We propose a novel Mixture-of-Scales (MoSs) architecture that substantially enhances the efficiency of late-stage refinement. The initial drafting stage, which is both fast and capable of producing high-quality drafts, remains unchanged. In contrast to existing next-scale prediction (VAR) models that employ sequential stages for refinement (see Fig. 4), we replace the multi-stage refiner with a single parallel refinement stage. The key components of our architecture include the *Frequency-aware Token Dispatcher*, the *MoSs Stage*, and *Cross-scale Self-Attention*.

Frequency-aware Token Dispatcher. After the drafting stage, shown in Fig. 4, the frequency-aware dispatcher assigns each spatially heterogeneous token region to the most suitable resolution scale for refinement. We utilize the local variance of the token embeddings to quantify the frequency of each token. Specifically, we define the frequency score $F(p)$ of a token p as the variance within a local spatial window $\mathcal{N}(p)$, averaged over all channels:

$$F(p) = \frac{1}{C}\sum_{c=1}^{C}(\mathbb{E}[I(q,c)^2] - (\mathbb{E}[I(q,c)])^2) \quad q \in \mathcal{N}(p) \tag{3}$$

where $\mathbb{E}[\cdot]$ denotes the average over all pixels q in the local window $\mathcal{N}(p)$ centered at p, and $I(q,c)$ is the value of pixel q in channel c. C is the number of channels in the token embedding. With the frequency score given, high-frequency tokens are dispatched to higher resolution scales for refinement, while low-frequency tokens are assigned to lower resolution scales.

To determine the frequency ranges that each parallel stage should process, we employ a top-down iterative heuristic based on binary search. The algorithm begins by assigning all tokens to the highest resolution scale and iteratively reallocates lower-frequency, less critical tokens to lower resolution scales using a grid search strategy. Throughout this process, evaluation metrics such as GenEval are used to maintain generation quality, and the assignment is performed offline to identify statistically appropriate percentiles for each frequency scale. Detailed algorithmic procedures are provided in the supplementary materials.

The MoSs Stage. At each resolution scale, only a subset of image tokens is selected for refinement. For instance, the top 25% highest-frequency tokens at the 64×64 scale, are chosen based on frequency criteria determined by the dispatcher. The MoSs stage aggregates all selected tokens across scales and inputs them as a unified sequence to the MoSs stage transformer, which jointly processes these

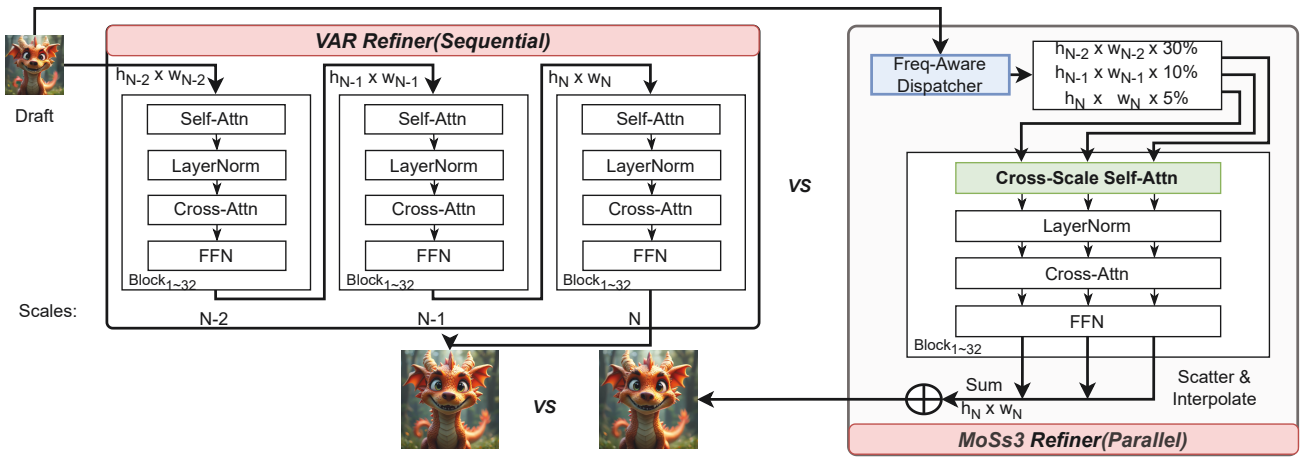


Figure 4: Standard VAR refiner versus MoSs.

mixed tokens and outputs the corresponding residual token sequence. Each token retains its positional embedding, indicating its location within the assigned image scale. Additionally, tokens at the MoSs stage have access to the drafter's KV cache, which provides essential contextual information from the drafting phase.

MoSs, breaking scale dependencies of the original VAR design, immediately brings in two benefits: (1) Mixed-scale refinement eliminates the requirement for sequential processing across scales, thereby reducing the number of computation-intensive steps needed by the conventional refiner; (2) Since only a small portion of the image is refined at each scale, the total number of refined tokens and the overall FLOPs are significantly reduced. MoSs is to some extent similar to the Mixture-of-Experts (MoE) approach, which gates sub-networks based on input semantics, but instead of activating sub-networks for all inputs, MoSs takes an alternative approach, activating only a portion of input tokens, mixing the tokens between different scales, and processing them with the full model.

Cross-Scale Self-Attention. The self-attention mechanism in the original VAR model enables each token to attend to all other tokens within the current scale. However, as the dispatcher aggressively prunes a substantial portion of tokens at each scale, we observe that pruning alone, while efficient, significantly degrades image quality. To address this, we introduce Cross-Scale Self-Attention (CSSA). In the CSSA framework, all token features across different resolution scales are jointly considered during self-attention computation. This enables tokens from various resolution stages to attend to one another, resulting in a unified self-attention space over the combined multi-scale token set. Consequently, tokens associated with low-frequency regions, although not directly processed at higher-resolution stages, can still transmit informative context to high-resolution tokens, thereby compensating for the potential information loss introduced by pruning. Notably, this approach remains training-free: CSSA reuses the original self-attention weights without any modification, simply extending the visibility range of tokens.

CSSA not only enhances generative quality but also improves computational efficiency. Instead of launching multiple self-attention kernels for each scale in the MoSs stage, CSSA allows combining tokens from different scales and processing them with a single attention kernel. To maximize CSSA's performance, we implement two specialized Triton kernels for positional embedding: one is designed to apply appropriate position embeddings of each token in the multi-scale token set, based on their assigned resolution scale, and the other precomputes positional embedding for all scales of different resolutions.

Experiments

Experimental Setup

Models and Evaluations. We evaluate our proposed MoSs framework on two representative next-scale prediction (VAR-based) models: **HART-0.7B** [Tang et al. 2024] and **Infinity-2B** [Han et al. 2024]. Both models generate high-resolution 1024×1024 images through a progressive autoregressive process. Our experiments focus on analyzing MoSs' impact on generation quality and inference efficiency. For image quality evaluation, we adopt two widely used benchmarks: **GenEval** [Ghosh, Hajishirzi, and Schmidt 2024] and **DPG** [Hu et al. 2024], which include both subjective and objective fidelity assessments across diverse prompts. To ensure reliable results, we run each setting with multiple random seeds and report averaged scores. Computational efficiency is measured by latency and memory usage on NVIDIA RTX 4090 (24GB) and NVIDIA A800 (80GB) GPUs, representing both consumer and data center hardware.

Implementation Details. Our MoSs implementation is model-agnostic and requires no retraining. In MoSs, we apply a heuristic frequency-aware strategy to allocate tokens to refinement stages based on local spatial frequency. For each MoS$s$$k$ configuration, we determine optimal frequency thresholds to assign tokens to k parallel refinement stages.

Method	Inference Efficiency				GenEval				DPG-Bench		
	#Scales	Latency↓	KVCache↓	SpeedUp↑	TwoObj↑	Counting↑	Colo_attr↑	Overall↑	Global↑	Relation↑	Overall↑
Infinity	13	1.29s	3.13GB	1	84.85	67.19	54.00	0.729	90.91	88.40	83.09
FastVAR	11	0.45s	1.42GB	2.87	81.21	67.23	52.39	0.721	86.93	89.57	82.52
+ MoSs2	11+2	0.63s	2.01GB	2.05	83.78	68.75	58.50	0.728	90.64	87.69	82.79
+ MoSs3	10+3	0.45s	1.23GB	2.87	83.59	68.44	54.50	0.724	87.65	89.33	82.68
+ MoSs4	9+4	0.38s	0.73GB	3.39	82.32	65.94	52.75	0.716	83.49	87.15	82.28
+ MoSs5	8+5	0.26s	0.45GB	**4.96**	80.87	64.53	51.25	0.705	89.72	88.49	82.38
HART	14	1.08s	2.79GB	1	51.26	30.25	17.75	0.498	84.71	86.47	78.99
FastVAR	14	0.72s	1.54GB	1.39	50.21	30.72	22.75	0.491	83.18	86.34	77.51
+ MoSs2	12+2	0.69s	0.90GB	**1.56**	50.91	30.69	21.50	0.488	82.76	86.63	77.49

Table 1: Quantitative evaluation of inference efficiency and generative quality across various methods. Latency is measured on an NVIDIA 4090 GPU with batch size 1, excluding VQVAE overhead shared by all methods.

Figure 5: Qualitative analysis of MoSs on the Infinity-2B model.

Specifically, for the infinity model, the percentile thresholds for token allocation are: [50, 25] for MoSs2, [50, 15, 5] for MoSs3, [80, 70, 30, 10] for MoSs4, and [90, 50, 15, 10, 5] for MoSs5. For the HART model, MoSs2 is [80,55]. For example, in MoSs2 with thresholds [50, 25], the last stage (highest resolution) refines tokens in the top $0 \sim 25\%$ frequency range (i.e., those with the highest local frequency), while the preceding stage handles tokens in the $25 \sim 50\%$ range. This assignment ensures that tokens corresponding to higher local frequencies are processed by stages with higher spatial resolution. All experiments are conducted on machines equipped with NVIDIA RTX 4090 (24GB) and NVIDIA A800 (80GB).

Main Results

Quality-Efficiency Trade-off. As shown in Tab. 1, MoSs achieves a favorable trade-off between generation quality and inference efficiency across all settings. On Infinity, MoSs delivers strong performance across all parallel configurations, achieving up to **2.05-4.96**× speedup with minimal quality degradation on GenEval and DPG benchmarks. Notably, while GenEval and DPG benchmarks slightly decline as more computation is skipped, the degradation remains within tolerable bounds. On HART, MoSs also achieves notable improvements, yielding a **1.56**× speedup and **67.74%** reduction in KV cache memory usage, but exhibits a slightly higher quality drop ($\sim 1\%$). We attribute this to HART's hybrid architecture that integrates both VAR and diffusion components. Since diffusion relies on progressively condi-

Method		GenEval			DPG-Bench			Inference Efficiency	
		TwoObj↑	Counting↑	Overall↑	Global↑	Relation↑	Overall↑	Latency↓	SpeedUp↑
Infinity	-	84.85	67.19	0.729	90.91	88.40	83.09	1.29	1
	RoPE	84.85	67.19	0.729	90.91	88.40	83.09	1.14	1.13
+MoSs2	SA	83.59	66.31	0.720	90.53	89.95	**82.80**	0.72	1.79
	CSSA	83.78	68.75	**0.728**	90.64	87.69	82.79	0.69	1.87
	CSSA+RoPE	83.78	68.75	0.728	90.64	87.69	82.79	0.63	**2.05**
+MoSs3	SA	82.43	66.38	0.715	88.90	86.04	**82.83**	0.54	2.39
	CSSA	83.59	68.44	**0.724**	87.65	89.33	82.68	0.49	2.63
	CSSA+RoPE	83.59	68.44	0.724	87.65	89.33	82.68	0.45	**2.87**
+MoSs4	SA	79.60	66.41	0.696	87.46	89.94	**82.44**	0.45	2.87
	CSSA	82.32	65.94	**0.716**	83.49	87.15	82.28	0.43	3.00
	CSSA+RoPE	82.32	65.94	0.716	83.49	87.15	82.28	0.38	**3.39**
+MoSs5	SA	74.31	60.80	0.674	86.95	89.92	81.94	0.31	4.16
	CSSA	80.87	64.53	**0.705**	89.72	88.49	**82.38**	0.29	4.45
	CSSA+RoPE	80.87	64.53	0.705	89.72	88.49	82.38	0.26	**4.96**

Table 2: Impact of Cross-Scale Self-Attention (CSSA) and RoPE in MoSs framework

tioned denoising steps, introducing parallel refinement disrupts its sequential generation process, leading to perceptual degradation. These results confirm that our frequency-based allocation achieves strong overall performance with minimal perceptual compromise.

Inference Speed and Memory Usage. We further analyze the inference latency and memory usage. On NVIDIA RTX 4090, for the Infinity model, MoSs reduces inference latency from a baseline of 1.27s to as low as 0.26s with the MoSs5 configuration, achieving a speedup of up to **4.96**×. Concurrently, the KV cache size decreased from **3.13 GB** to **0.45 GB**. For the HART model, under the MoSs2 configuration, inference latency was reduced by **1.56**× compared to the baseline, with latency decreasing from 1.08s to 0.69s. Meanwhile, the KV cache size was significantly reduced from **2.79 GB** to **0.90 GB**. Furthermore, on a server-grade NVIDIA A800 (80GB) GPU, MoSs demonstrates 1.64-2.50× acceleration for the Infinity model and 1.39× acceleration for the HART model. Notably, the reported speedup ratios are based on the backbone model, with the decoder time overhead for both the Infinity and HART models being approximately 0.25s.

Qualitative Results. Fig. 5 showcases the visual outputs generated by the original Infinity model and its various MoSs variants. As illustrated, MoSs maintains high visual fidelity while providing substantial acceleration and a reduction in KV cache usage.

Efficiency Analysis

Cross-scale Self-Attention. Experimental results presented in Tab. 2 demonstrate that, on the Infinity model, CSSA consistently improves quality while enhancing efficiency. Across all MoSs2 $\sim$ MoSs5 configurations, CSSA outperforms standard self-attention (SA) on GenEval metrics and maintains comparable or better results on DPG-Bench. More importantly, it achieves lower latency and higher speedup across the board. For example, in MoSs5, GenEval-Overall improves from 0.674 to 0.705, and speedup increases from $4.16\times$ to $4.45\times$. These findings indicate that CSSA serves as a vital component for efficient and high-fidelity refinement in the MoSs pipeline.

RoPE2D. To further improve inference efficiency, we develop a Triton-based implementation of RoPE2D. The experimental results show that with the addition of the optimized kernel to CASS, the inference is accelerated by $2.05\times$ to $4.96\times$ overall. This optimization ensures that the CSSA module remains computationally efficient and well-suited for practical deployment.

Conclusion and Future Work

We propose **M**ixture **o**f **S**cale**s** (**MoSs**), a lightweight, training-free framework for efficient high-resolution autoregressive image generation. MoSs addresses refinement inefficiencies through a parallel multi-scale strategy that incorporates frequency-aware token dispatch and cross-scale self-attention. On the state-of-the-art Infinity model, MoSs achieves **2.05-4.96× speedup**, up to **85.62% KV cache reduction**, incurring only **0.1-2.4%** loss on GenEval quality. Our method delivers substantial efficiency gains with minimal impact on image fidelity, enabling the practical deployment of large-scale autoregressive models. While MoSs provides strong performance benefits on high-resolution VAR networks, we observe that, for low-resolution VAR networks or hybrid models that combine VAR with diffusion models, directly applying training-free MoSs presents certain challenges with some degradation in image quality. We believe that with appropriate fine-tuning, both the accuracy and speed of the model can be further improved, enhancing its applicability to a wider range of architectures and scenarios.

Acknowledgments

This work was sponsored by the Shanghai Rising-Star Program (No. 24QB2706200) and the National Natural Science Foundation of China (No. U21B2031).

References

Bolya, D.; Fu, C.-Y.; Dai, X.; Zhang, P.; Feichtenhofer, C.; and Hoffman, J. 2022. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*.

Chang, H.; Zhang, H.; Jiang, L.; Liu, C.; and Freeman, W. T. 2022. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11315–11325.

Chen, J.; Ge, C.; Xie, E.; Wu, Y.; Yao, L.; Ren, X.; Wang, Z.; Luo, P.; Lu, H.; and Li, Z. 2024a. Pixart-σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. In *European Conference on Computer Vision*, 74–91. Springer.

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

Chen, X.; Mishra, N.; Rohaninejad, M.; and Abbeel, P. 2018. Pixelsnail: An improved autoregressive generative model. In *International conference on machine learning*, 864–872. PMLR.

Chen, Z.; Ma, X.; Fang, G.; and Wang, X. 2024b. Collaborative Decoding Makes Visual Auto-Regressive Modeling Efficient. *arXiv preprint arXiv:2411.17787*.

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

Esser, P.; Rombach, R.; and Ommer, B. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12873–12883.

Fan, L.; Li, T.; Qin, S.; Li, Y.; Sun, C.; Rubinstein, M.; Sun, D.; He, K.; and Tian, Y. 2024. Fluid: Scaling autoregressive text-to-image generative models with continuous tokens. *arXiv preprint arXiv:2410.13863*.

Gao, J.; Liu, W.; Sun, W.; Wang, S.; Song, X.; Shang, T.; Chen, S.; Li, H.; Yang, X.; Yan, Y.; et al. 2025. Mars: Mesh autoregressive model for 3d shape detailization. *arXiv preprint arXiv:2502.11390*.

Ghosh, D.; Hajishirzi, H.; and Schmidt, L. 2024. GenEval: An object-focused framework for evaluating text-to-image alignment. 36.

Gregor, K.; Danihelka, I.; Mnih, A.; Blundell, C.; and Wierstra, D. 2014. Deep autoregressive networks. In *International Conference on Machine Learning*, 1242–1250. PMLR.

Guo, H.; Li, Y.; Zhang, T.; Wang, J.; Dai, T.; Xia, S.-T.; and Benini, L. 2025. FastVAR: Linear Visual Autoregressive Modeling via Cached Token Pruning. *arXiv preprint arXiv:2503.23367*.

Han, J.; Liu, J.; Jiang, Y.; Yan, B.; Zhang, Y.; Yuan, Z.; Peng, B.; and Liu, X. 2024. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. *arXiv preprint arXiv:2412.04431*.

Henighan, T.; Kaplan, J.; Katz, M.; Chen, M.; Hesse, C.; Jackson, J.; Jun, H.; Brown, T. B.; Dhariwal, P.; Gray, S.; et al. 2020. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*.

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

Hu, X.; Wang, R.; Fang, Y.; Fu, B.; Cheng, P.; and Yu, G. 2024. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*.

Jang, D.; Park, S.; Yang, J. Y.; Jung, Y.; Yun, J.; Kundu, S.; Kim, S.-Y.; and Yang, E. 2024. Lantern: Accelerating visual autoregressive models with relaxed speculative decoding. *arXiv preprint arXiv:2410.03355*.

Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T. B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; and Amodei, D. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Li, T.; Chang, H.; Mishra, S.; Zhang, H.; Katabi, D.; and Krishnan, D. 2023. Mage: Masked generative encoder to unify representation learning and image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2142–2152.

Li, T.; Sun, Q.; Fan, L.; and He, K. 2025. Fractal generative models. *arXiv preprint arXiv:2502.17437*.

Li, T.; Tian, Y.; Li, H.; Deng, M.; and He, K. 2024a. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37: 56424–56445.

Li, X.; Qiu, K.; Chen, H.; Kuen, J.; Lin, Z.; Singh, R.; and Raj, B. 2024b. Controlvar: Exploring controllable visual autoregressive modeling. *arXiv preprint arXiv:2406.09750*.

Li, Z.; Cheng, T.; Chen, S.; Sun, P.; Shen, H.; Ran, L.; Chen, X.; Liu, W.; and Wang, X. 2024c. Controlar: Controllable image generation with autoregressive models. *arXiv preprint arXiv:2410.02705*.

Liang, Y.; Ge, C.; Tong, Z.; Song, Y.; Wang, J.; and Xie, P. 2022. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

Ma, X.; Zhou, M.; Liang, T.; Bai, Y.; Zhao, T.; Chen, H.; and Jin, Y. 2024. Star: Scale-wise text-to-image generation via auto-regressive representations. *arXiv preprint arXiv:2406.10797*.

Parmar, N.; Vaswani, A.; Uszkoreit, J.; Kaiser, L.; Shazeer, N.; Ku, A.; and Tran, D. 2018. Image transformer. In *International conference on machine learning*, 4055–4064. PMLR.

Qiu, K.; Li, X.; Chen, H.; Sun, J.; Wang, J.; Lin, Z.; Savvides, M.; and Raj, B. 2024. Efficient autoregressive audio modeling via next-scale prediction. *arXiv preprint arXiv:2408.09027*.

Qu, Y.; Yuan, K.; Hao, J.; Zhao, K.; Xie, Q.; Sun, M.; and Zhou, C. 2025. Visual Autoregressive Modeling for Image Super-Resolution. *arXiv preprint arXiv:2501.18993*.

Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; and Sutskever, I. 2021. Zero-shot text-to-image generation. In *International conference on machine learning*, 8821–8831. Pmlr.

Rao, Y.; Zhao, W.; Liu, B.; Lu, J.; Zhou, J.; and Hsieh, C.-J. 2021. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34: 13937–13949.

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

Rhee, H.; Jang, Y. I.; Kim, S.; and Cho, N. I. 2022. LC-FDNet: Learned lossless image compression with frequency decomposition network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6033–6042.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Shukor, M.; Fini, E.; da Costa, V. G. T.; Cord, M.; Susskind, J.; and El-Nouby, A. 2025. Scaling Laws for Native Multimodal Models Scaling Laws for Native Multimodal Models. *arXiv preprint arXiv:2504.07951*.

Siddeq, M. M.; and Rodrigues, M. A. 2017. A novel high-frequency encoding algorithm for image compression. *EURASIP Journal on Advances in Signal Processing*, 2017: 1–17.

Tang, H.; Wu, Y.; Yang, S.; Xie, E.; Chen, J.; Chen, J.; Zhang, Z.; Cai, H.; Lu, Y.; and Han, S. 2024. Hart: Efficient visual generation with hybrid autoregressive transformer. *arXiv preprint arXiv:2410.10812*.

Teng, Y.; Shi, H.; Liu, X.; Ning, X.; Dai, G.; Wang, Y.; Li, Z.; and Liu, X. 2024. Accelerating auto-regressive text-to-image generation with training-free speculative jacobi decoding. *arXiv preprint arXiv:2410.01699*.

Tian, K.; Jiang, Y.; Yuan, Z.; Peng, B.; and Wang, L. 2024. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37: 84839–84865.

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

Van Den Oord, A.; Kalchbrenner, N.; and Kavukcuoglu, K. 2016. Pixel recurrent neural networks. In *International conference on machine learning*, 1747–1756. PMLR.

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

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

Wang, X.; Zhang, X.; Luo, Z.; Sun, Q.; Cui, Y.; Wang, J.; Zhang, F.; Wang, Y.; Li, Z.; Yu, Q.; et al. 2024. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*.

Xiao, G.; Tian, Y.; Chen, B.; Han, S.; and Lewis, M. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

Xie, R.; Zhao, T.; Yuan, Z.; Wan, R.; Gao, W.; Zhu, Z.; Ning, X.; and Wang, Y. 2024. LiteVAR: Compressing Visual Autoregressive Modelling with Efficient Attention and Quantization. *arXiv preprint arXiv:2411.17178*.

Zhang, J.; Xiong, F.; and Xu, M. 2024. G3pt: Unleash the power of autoregressive modeling in 3d generation via cross-scale querying transformer. *arXiv preprint arXiv:2409.06322*.

Zhang, Q.; Dai, X.; Yang, N.; An, X.; Feng, Z.; and Ren, X. 2024. Var-clip: Text-to-image generator with visual autoregressive modeling. *arXiv preprint arXiv:2408.01181*.

Zhuang, X.; Xie, Y.; Deng, Y.; Liang, L.; Ru, J.; Yin, Y.; and Zou, Y. 2025. VARGPT: Unified Understanding and Generation in a Visual Autoregressive Multimodal Large Language Model. *arXiv preprint arXiv:2501.12327*.