This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

LazyVAR: Accelerating Visual Autoregressive Models via Scale-wise Token Pruning and Parallel Group Decoding

Rongge Mao[1,2] Chengqi Dong[1,2] S Kevin Zhou[1,2,3,4*]
[1]School of Biomedical Engineering, Division of Life Sciences and Medicine, USTC
[2]MIRACLE Center, Suzhou Institute for Advanced Research, USTC
[3]Key Laboratory of Intelligent Information Processing of CAS, ICT, CAS
[4]State Key Laboratory of Precision and Intelligent Chemistry, USTC
ronggemao@mail.ustc.edu.cn dongcq@mail.ustc.edu.cn skevinzhou@ustc.edu.cn

Abstract

Visual Autoregressive (VAR) modeling introduces a new paradigm for image generation by extending autoregressive mechanisms from next-token prediction to next-scale prediction, achieving remarkable performance. However, as the number of tokens increases rapidly with scale, processing full token maps at high resolution becomes computationally expensive. In addition, the inherently sequential nature of autoregressive modeling prevents parallel inference across scales, which further increases latency. To address these challenges, we propose ***LazyVAR****, a training-free and plug-and-play acceleration method for VAR models. Our key observation is that the similarity of aggregated latent features between adjacent scales progressively increases with the scale index, reaching particularly higher values at larger scales. We treat this similarity as a* ***Scale-wise Update Index****, which serves as the pruning criterion. Consequently, more tokens can be pruned at larger scales to improve efficiency. Furthermore, we propose* ***Parallel Group Decoding****, which leverages this high similarity at larger scales to decode tokens from different scales in parallel, further accelerating inference. Experimental results show that the proposed LazyVAR achieves up to a* ***2.94****× speedup of recent VAR models with negligible performance degradation, allowing the Infinity-2B text-to-image model to generate 1024×1024 resolution images in 0.5 seconds on a single RTX 4090 GPU.*

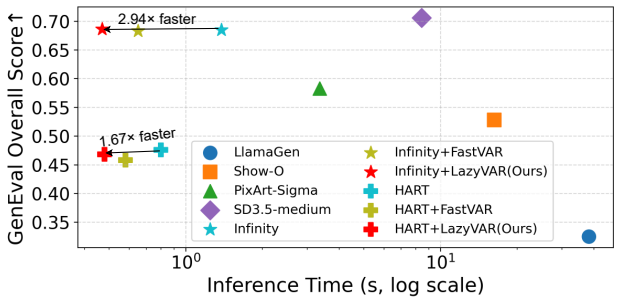
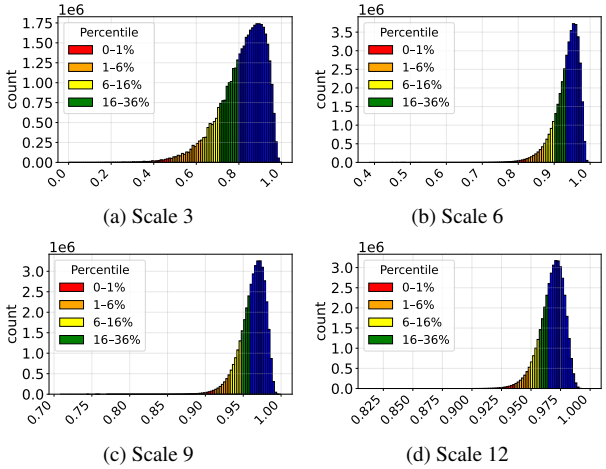


Figure 1. LazyVAR can achieve a noticeable speedup with negligible performance drop.



(a) Scale 3 (b) Scale 6

(c) Scale 9 (d) Scale 12

Figure 2. Distribution of the cosine similarity between adjacent aggregated latent features across scales.

1. Introduction

In recent years, diffusion-based models have emerged as the dominant paradigm in image generation [14, 27, 44, 46, 51]. Despite their remarkable success, an alternative line of research based on autoregressive (AR) generation [15, 54, 61] has shown a level of generation quality that is comparable to, and in some benchmarks even surpasses, that

*Corresponding Author

of diffusion-based methods. Moreover, AR models offer several distinct advantages, including a unified generative formulation [56, 66], compatibility with advanced training techniques developed for large-scale language models [1, 59], and strong scalability [25, 29].

Among recent autoregressive frameworks, the VAR [58] model has attracted particular attention for reformulating the autoregressive paradigm from next-token prediction to next-scale prediction, achieving superior visual fidelity and semantic consistency. However, due to the inherently sequential nature of autoregression, the decoding steps in VAR cannot be executed in parallel. Moreover, as the scale expands, the number of tokens increases quadratically with respect to n, resulting in $\mathcal{O}(n^2)$ complexity growth, while the attention computation introduces an even higher $\mathcal{O}(n^4)$ complexity, significantly inflating the overall computational cost. Prior work such as FastVAR [21] adopts a frequency-domain-based pruning strategy to accelerate VAR models. In contrast, our approach leverages the model's token update dynamics to determine pruning decisions. Furthermore, we exploit our observations to group the pruned steps for parallel inference, thereby achieving further acceleration.

In the VAR framework, a latent space variable $\hat{f}(h, w)$ is maintained to aggregate the residual predictions across all scales. We refer to this variable as the aggregated latent feature. Previous studies seeking to interpret VAR have primarily focused on two aspects: decoding the aggregated latent feature at various steps [21] into the pixel space using a VQVAE [61] decoder, and analyzing attention patterns in VAR blocks [33], which are based on the Transformer architecture [62]. In contrast, our study focuses on investigating the similarity of $\hat{f}$ between adjacent scales. Motivated by the key findings (see Section 3.2 for more details), we design **LazyVAR**, which integrates two complementary strategies to enhance inference efficiency: (1) At larger scales, LazyVAR determines whether each token should be activated based on its update behavior in the previous scale, thereby reducing the number of active tokens. (2) It further enables multiple steps that are originally non-parallel across scales to be grouped for parallel decoding. Experimental results show that LazyVAR accelerates the original Infinity model by **2.94**$\times$ with negligible degradation in generation quality.

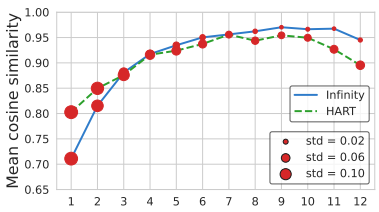
2. Related Work

Autoregressive Image Generation. Early works [7, 60] performed autoregressive generation in the pixel space. Subsequent methods [45, 61] such as VQVAE introduced vector quantization to learn discrete latent representations, and VQGAN [15] extended this framework using Transformer architectures. MaskGIT [4] proposed a masked modeling strategy analogous to BERT [13], and the MAR [36] framework unified autoregressive (AR) and masked image modeling (MIM) [2, 5, 23, 35, 71], referring to it as masked autoregressive modeling and highlighting the crucial role of bidirectional attention. Later works, including RAR [74] and RandAR [43], explored random-order token generation, enabling partial parallelization during inference. Unlike patch-level autoregressive models, VAR proposed predicting the next scale, while FAR [73] adopted a similar principle for frequency-domain generation. Nonetheless, VAR models still face the challenge of handling a large number of tokens at larger scales.

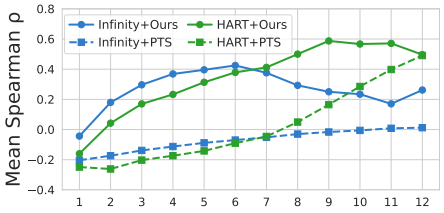
The VAR Family. The original VAR paper demonstrated strong class-conditional generation performance on ImageNet [47], inspiring a series of subsequent extensions. Infinity [22] introduced a discrete autoencoder with an infinitely large codebook size, while HART [55] investigated hybrid discrete and continuous tokenization schemes. Another variant, MVAR [76], formulated next-scale prediction as a Markov process. Further developments have expanded the VAR family: AREdit [67], VARIN [10], and EditInfinity [64] enabled training-free image editing; CycleVAR [39] adapted the framework for image-to-image translation; ScaleKV [33] optimizes Key-Value (KV) cache allocation to reduce memory consumption during inference; and CoDE [8] achieved accelerated generation by combining VAR models of different sizes in cooperative inference. FastVAR [21] introduced *Pivotal Token Selection* (PTS) to prune redundant tokens based on the deviation between individual token norms and the average token norm, as well as *Cached Token Restoration* (CTR) to preserve spatial coherence. SparseVAR [9] distinguishes high- and low-frequency regions based on the observation that different blocks in next-scale prediction models tend to focus on distinct regions. SkipVAR [32] selectively skips certain scales or CFG branches by training an additional classifier.

Efficient Image Generation. The widespread adoption of diffusion models has stimulated extensive research on accelerating their inference [3, 18, 34, 40, 63, 75, 77]. Some studies have applied commonly used deep learning acceleration techniques, such as model pruning [63], quantization [20, 37, 50], and knowledge distillation [42, 48], to diffusion models. Other approaches exploit the intrinsic properties of diffusion models. For example, DDIM [52] accelerates sampling by generating intermediate denoised images in a non-Markovian manner, and DeepCache [41] reduces computational cost by reusing high-level semantic representations across adjacent denoising steps. In a similar spirit, our work aims to exploit the intrinsic properties of VAR models to accelerate the generation process, instead of depending on general-purpose deep learning acceleration methods. Nevertheless, due to fundamental architectural and generative differences, acceleration methods developed for diffusion models cannot be directly applied to VAR.

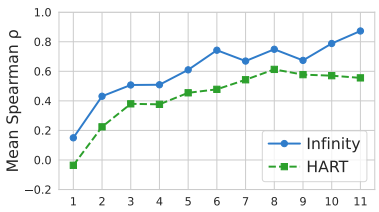
In parallel, the autoregressive modeling community,



(a) Mean and standard deviation of $c_i(h, w)$ as a function of scale.



(b) Spearman correlation between the pruning criterion and the patch-wise pixel variance across scales.



(c) Spearman correlation between $c_i(h, w)$ and $c_{i+1}(h, w)$.

Figure 3. Investigation of the cosine similarity of aggregated latent features between adjacent scales in two families of VAR-based text-to-image models, Infinity [22] and HART [55]. The text prompts are randomly sampled from a publicly available Midjourney v6 prompt dataset, with a total of 10,000 samples. In subfigure (b), Ours employs the cosine similarity of adjacent aggregated latent features as the pruning criterion, whereas PTS follows the approach proposed in FastVAR [21].

particularly in the context of large language models (LLMs), has developed a diverse range of acceleration techniques [17, 24, 28, 30, 49, 57, 69]. However, these approaches are designed for next-token prediction rather than next-scale prediction, limiting their applicability to VAR. Efforts to accelerate MAR-like models, such as ReCAP [38] and LazyMAR [72], integrate KV-cache into non-causal attention mechanisms to reduce computation. In contrast, VAR already utilizes the KV-cache effectively, necessitating novel strategies specifically tailored to its hierarchical and scale-autoregressive generative paradigm.

3. Method

3.1. Preliminary

The VAR framework reformulates conventional autoregressive generation by shifting the prediction target from the *next token* to the *next scale*. Instead of generating a single token at each step, VAR operates on multi-scale token maps, where each autoregressive unit corresponds to a specific spatial resolution level.

Given an image $\mathbf{I} \in \mathbb{R}^{3\times H\times W}$, VAR first employs a visual tokenizer to quantize it into K scales token maps $\mathcal{R} = \{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_K\}$, where the resolution of $\mathbf{r}_k$ is (h_k, w_k) and $(h_1, w_1) < (h_2, w_2) < \cdots < (h_\mathrm{K}, w_\mathrm{K})$. These token maps are arranged in a coarse-to-fine hierarchy, enabling the model to learn global structure at smaller scales and refine visual details at larger scales.

Formally, the joint distribution of the multi-scale token maps is modeled autoregressively as:

$$p(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_\mathrm{K}) = \prod_{k=1}^{K} p(\mathbf{r}_k \mid \mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{k-1}, c), \quad (1)$$

where c denotes either the class-conditional information or the text embeddings obtained from a pretrained language model given a textual prompt.

To facilitate stable learning, VAR adopts a residual formulation that decomposes the final latent representation into contributions from multiple scales. The aggregated latent feature at scale i can be expressed as the interpolated cumulative sum of the preceding scales output:

$$\hat{f}_i = \sum_{k=1}^{i} \mathrm{Interpolate}(r_k, (h_\mathrm{K}, w_\mathrm{K})), \quad (2)$$

where $\mathrm{Interpolate}(\cdot)$ denotes the upsampling operator that aligns the spatial resolution of the lower-scale features to the final scale $(h_\mathrm{K}, w_\mathrm{K})$ before aggregation. This hierarchical prediction mechanism enables the model to progressively integrate coarse semantic and structural information with fine-grained spatial details.

During inference, although VAR can benefit from the KV-cache, its autoregressive nature prevents different steps from being executed in parallel. In addition, each step still requires processing all $h_k \times w_k$ tokens at the corresponding scale. As the spatial resolution increases with the scale, the overall computational cost grows substantially.

3.2. Key Findings and Motivation

We compute $c_i(h, w) = cos\langle \hat{f}_{i-1}(h, w), \hat{f}_i(h, w)\rangle$. Analysis of this metric reveals key findings regarding the behavior of aggregated latent features across adjacent scales, motivating our approach.

Finding 1. Similarity of adjacent aggregated latent features progressively increases across scales . As shown in Figure 2, the distribution of $c_i(h, w)$ exhibit a right-skewed, long-tailed pattern at all scales (the first scale, denoted as Scale 1). With increasing scale, the entire distribution gradually shifts toward higher similarity values, indicating a progressive homogenization of adjacent aggregated latent features. At larger scales, the distribution becomes sharply peaked near 1.0, with only a small fraction of tokens residing in the lower-similarity tail. For instance, at the eleventh scale, approximately 94% of tokens achieve a cosine similarity greater than 0.95.

Finding 2. The model is aware of high-frequency details and actively updates the corresponding tokens. As

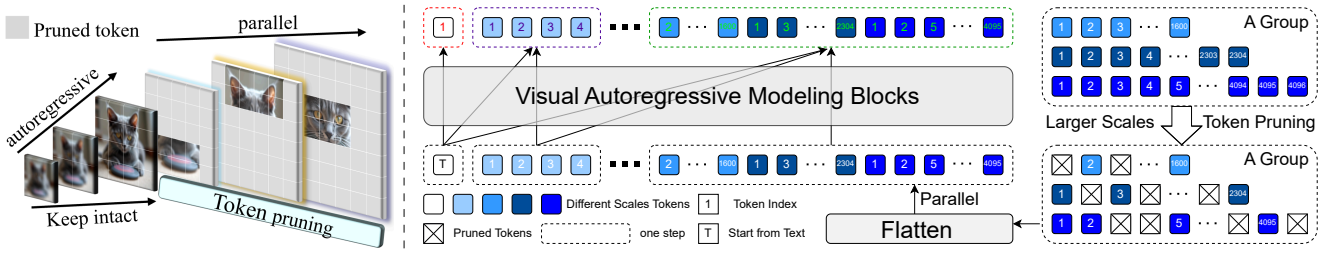


Figure 4. **Overview of LazyVAR.** The smaller scales are kept intact to preserve the semantic and structural integrity of the generated image. At larger scales, pruning is guided by the Update_Index, and the pruned steps are grouped to enable parallel inference.

shown in Figure 3b, $c_i(h, w)$ exhibits a significant Spearman correlation [53] with the variance $\sigma(h, w)$ of the corresponding patches in the final generated image; specifically, we compute $\rho_i = \text{Spearman}\langle -c_i(h, w), \sigma(h, w)\rangle$. This finding indicates that, at larger scales, tokens undergoing frequent updates tend to correspond to high-frequency details in the generated image, while the model preserves the tokens associated with the already well-generated background regions.

Finding 3. Update patterns remain consistent across adjacent scales. We further compute $s_i = \text{Spearman}\langle c_i(h, w), c_{i+1}(h, w)\rangle$ to investigate whether the model exhibits cross-scale consistency in its token update patterns. As depicted in Figure 3c, there exists a significant Spearman correlation between $c_i(h, w)$ and $c_{i+1}(h, w)$. This indicates that tokens experiencing substantial updates at scale i tend to undergo similarly large updates at scale $i + 1$. Such behavior reveals the consistency of the model's update dynamics, suggesting that the model effectively localizes tokens associated with fine-grained details and iteratively refines them across multiple scales.

Motivation 1. As shown in Finding 1, at larger scales, after incorporating the residuals from the current-scale predictions, many tokens within the aggregated latent feature exhibit very high cosine similarity, indicating that their updates are relatively small. This suggests that a considerable portion of token forward computations can be skipped without significantly affecting the overall representation at larger scales.

Motivation 2. As discussed in Finding 2 and Finding 3, the cosine similarity between aggregated latent features of adjacent scales shows a clear correlation across scales and is also related to the frequency characteristics of the corresponding image patches. This property makes it a reliable criterion for token pruning. The model can therefore automatically identify which tokens correspond to high-frequency details and should be prioritized for updates.

Motivation 3. Since the aggregated latent feature at each scale is downsampled to initialize the next scale, the requirement to update the aggregated latent feature at every step prevents parallel inference across scales. As shown in Finding 1, given that the adjacent aggregated latent features are highly similar at larger scales, the unupdated aggregated latent feature can be downsampled to generate multiple scale inputs. Consequently, multiple scales can be decoded in parallel, thereby improving efficiency.

3.3. LazyVAR

We propose LazyVAR, a method that combines Update Index Guided Token Pruning (UIGTP) and Parallel Group Decoding (PGD) to accelerate inference speed for VAR models. The core idea is to perform token pruning and parallelize multiple pruned steps within a group.

Scale-wise Update Index Guided Token Pruning. Formally, the aggregated latent feature has been defined in Equation (2), and it can be expressed recursively as

$$\hat{f}_k = \hat{f}_{k-1} + \text{Interpolate}\big(r_k, (h_\text{K}, w_\text{K})\big), \tag{3}$$

where r_k denotes the residual predictions at scale k, and (h_K, w_K) corresponds to the largest spatial resolution.

To quantify the scale-wise updates of each token in the aggregated latent feature, we resort to the token-wise cosine similarity between adjacent aggregated latent features:

$$\text{Update_Index}_k = 1 - \cos\langle \hat{f}_{k-1},\ \hat{f}_k\rangle \in [0, 2]^{h_\text{K} \times w_\text{K}}. \tag{4}$$

Tokens exhibiting low cosine similarity values indicate substantial feature updates at the current scale. At larger scales, a considerable portion of tokens in the aggregated latent feature undergoes only minor changes. Consequently, these tokens can be omitted from the transformer input, substantially reducing the number of active tokens and improving computational efficiency. Furthermore, by analyzing the Spearman correlation between $\text{Update_Index}_{k-1}$ and Update_Index_k, we observe that tokens showing small updates at one scale tend to exhibit similarly small updates at the subsequent scales. These observations suggest that $\text{Update_Index}_{k-1}$ can serve as a reliable criterion for token pruning at scale k.

Parallel Group Decoding. To further enhance inference efficiency, we propose grouping multiple adjacent steps together for parallel decoding. Specifically, each group consists of p successive steps beginning at scale k. The original

Table 1. Comparison of inference efficiency and generation quality on GenEval and MJHQ-30K benchmarks. All models were evaluated on a single RTX 4090 GPU. ↑ indicates higher is better, and ↓ indicates lower is better. †Models marked with a dagger can only generate images up to 512×512 resolution; all others support 1024×1024 resolution.

		Inference Efficiency						GenEval	MJHQ-30K		
Model	**Category**	**Steps**	**Params.**	**Time↓**	**Speed-up↑**	**FLOPs↓**	**Memory↓**	**Overall↑**	**FID↓**	**FID*↓**	**CLIP↑**
LlamaGen† [54]	AR	1024	0.8B	38.27 s	–	–	5.0 GB	0.325	–	–	–
Show-O† [70]	MAR	50	1.3B	16.25 s	–	–	7.4 GB	0.528	–	–	–
PixArt-Σ [6]	Diffusion	20	0.6B	3.36 s	–	–	13.9GB	0.583	6.55	–	0.283
SD3.5-M [16]	Diffusion	40	2.5B	8.44 s	–	–	19.9GB	0.706	8.48	–	0.281
Infinity [22]	VAR	13	2.0B	1.38 s	1.00×	101.58T	16.1GB	0.685	9.80	0.00	0.277
+FastVAR [21]	VAR	11	2.0B	0.65 s	2.12×	44.15T	11.9GB	0.683	10.05	2.03	**0.279**
+LazyVAR	VAR	**10**	2.0B	**0.47 s**	**2.94**×	**34.32T**	**11.2GB**	**0.686**	**9.83**	**1.46**	0.277
HART [55]	VAR	14	0.7B	0.80 s	1.00×	45.59T	20.6GB	0.476	10.98	0.00	0.283
+FastVAR [21]	VAR	14	0.7B	0.58 s	1.38×	34.11T	20.5GB	0.458	**10.54**	8.06	0.280
+LazyVAR	VAR	**13**	0.7B	**0.48 s**	**1.67**×	**32.09T**	**20.2GB**	**0.468**	11.04	**3.77**	**0.281**

input for scale k is defined as

$$\tilde{r}_k = \text{Interpolate}\big(\hat{f}_{k-1}, (h_k, w_k)\big). \tag{5}$$

We define the approximated inputs for the subsequent scales within the same group as

$$\tilde{r}_{k+m} \approx \tilde{r'}_{k+m} = \text{Interpolate}\big(\hat{f}_{k-1}, (h_{k+m}, w_{k+m})\big), \tag{6}$$

When the aggregated latent features within a group are highly similar, that is,

$$\hat{f}_{k-1} \approx \hat{f}_{k+m-1}, \quad \forall m \in [0, p-1], \tag{7}$$

and the downsampling operation further reduces such differences, the above approximation $\tilde{r}_{k+m} \approx \tilde{r'}_{k+m}$ holds.

Originally, the decoding process across different scales is sequential, and the residual prediction at scale k is computed through stacked transformer blocks as

$$r_k = \text{Blocks}\big(\tilde{r}_1, \tilde{r}_2, \ldots, \tilde{r}_k, c\big). \tag{8}$$

By approximating all interpolated inputs within the group using $\tilde{r'}_k$, the decoding of all scales inside the group can be performed in parallel:

$$r_k, \ldots, r_{k+p} = \text{Blocks}\big(\tilde{r}_1, \tilde{r}_2, \ldots, \tilde{r'}_k, \ldots, \tilde{r'}_{k+p}, c\big). \tag{9}$$

This strategy enables the p subsequent scales to be decoded in parallel rather than sequentially, effectively reducing computational latency without degrading image generation quality.

Implementation of LazyVAR. Since the aggregated latent features undergo larger updates at smaller scales, and these scales are crucial for determining the structural and semantic content of the generated image while containing relatively few tokens, pruning them yields only marginal acceleration and leads to a noticeable degradation in generation quality. Therefore, we keep the inference process at smaller scales intact. Specifically, pruning starts from scale i, and the subsequent $p-1$ steps are grouped together, where each scale k retains n_k tokens after pruning. To avoid updating the same token position across different scales within the same group, we assign the tokens with the highest Update-Index values to the largest scale, while tokens with relatively lower Update-Index values are assigned to smaller scales. For scale i, we compute the cumulative number of retained tokens across scales $i+1$ to $i+p$ as follows:

$$\text{cum}_i = \sum_{k=i+1}^{i+p} n_k \cdot \big(h_\text{K} \times w_\text{K}\big) / \big(h_k \times w_k\big) \tag{10}$$

For scale $i+m$, $\text{Update_Index}_{i-1}$ is first downsampled to (h_{i+m}, w_{i+m}), and then the indices of n_{i+m} tokens starting from the cum_{i+m}-th largest are selected as the retained token indices, as illustrated in Equation (11). After the pruned tokens are output from the blocks, their spatial positions are restored based on the retained token indices. Since the $\text{Update_Index}_{k-1}$ values of the pruned tokens are very small, we set the residual predictions at these token positions for scale k to zero.

4. Experiment

4.1. Experimental Setup

Models and Evaluations. We conducted experiments using two trained VAR-based text-to-image generation models, namely Infinity [22] and HART [55]. To ensure a fair comparison, all hyperparameters unrelated to this study were fixed to their default settings during inference. The GenEval [19] benchmark was employed to evaluate the semantic consistency of generated images, while the MJHQ-30K [31] benchmark was utilized to assess the perceptual quality of the images. Additionally, the HPS v2.1 [68]

$$\mathcal{I}_{i+m} = \mathrm{Sort}^{\downarrow}\Big(\mathrm{Interpolate}\big(\mathrm{Update_Index}_{i-1}, (h_{i+m}, w_{i+m})\big)\Big)\big[\mathrm{cum}_{i+m} : \mathrm{cum}_{i+m} + n_{i+m}\big], \quad \forall m \in [0, p-1]. \quad (11)$$

Table 2. Comprehensive quantitative evaluation on the GenEval benchmark.

Method	Speed-up↑	Acc_i.	Acc_p.	Single.	Two.	Colors	Col_att.	Position	Counting	Average
PixArt-Σ [6]	–	56.87%	74.32%	98.44%	66.41%	80.59%	35.00%	18.00%	51.56%	58.33%
SD3.5-M [16]	–	69.76%	84.63%	98.44%	85.10%	84.84%	62.25%	31.75%	61.25%	70.61%
Infinity [22]	1.00×	67.41%	80.11%	99.69%	83.84%	81.12%	54.00%	29.00%	63.44%	68.51%
+FastVAR [21]	2.12×	67.31%	79.57%	99.38%	**82.58%**	**81.38%**	55.75%	**30.00%**	60.94%	68.34%
+LazyVAR	**2.94**×	**67.59%**	**80.83%**	**99.69%**	82.07%	80.32%	**57.25%**	**30.00%**	**62.50%**	**68.64%**
HART [55]	1.00×	46.02%	65.28%	94.06%	48.74%	78.46%	17.25%	11.75%	35.31%	47.60%
+FastVAR [21]	1.38×	44.21%	63.47%	91.88%	45.96%	76.33%	15.75%	**11.00%**	**33.75%**	45.78%
+LazyVAR	**1.67**×	**45.30%**	**64.92%**	**94.38%**	**47.22%**	**78.72%**	**18.50%**	10.75%	31.25%	**46.80%**

Table 3. Quantitative comparisons of FID and CLIP scores on several categories in the MJHQ-30K benchmark.

Methods	logo		people		plants		vehicles	
	FID↓	CLIP↑	FID↓	CLIP↑	FID↓	CLIP↑	FID↓	CLIP↑
SD3.5-M	31.88	0.287	29.57	0.283	29.71	0.274	22.91	0.280
PixArt-Σ	27.53	0.274	26.18	0.289	22.95	0.276	21.80	0.283
Infinity	29.79	0.276	30.52	0.281	29.72	0.265	24.76	0.276
FastVAR	31.63	0.277	31.06	0.284	30.38	0.266	24.77	0.278
LazyVAR	30.51	0.275	30.52	0.282	29.97	0.265	24.23	0.275
HART	30.19	0.280	30.60	0.285	30.73	0.279	29.64	0.276
FastVAR	42.51	0.269	28.03	0.284	36.66	0.274	26.50	0.277
LazyVAR	36.49	0.274	28.60	0.284	31.23	0.275	26.75	0.274

benchmark, which aligns more closely with human preferences in image generation, was adopted to further validate the quality of our method. For efficiency evaluation, we report several key metrics, including the inference time (excluding the time required for processing text prompts using the pre-trained language model), the total forward FLOPs, the number of model parameters, the number of sampling steps, the acceleration ratio compared to the original model, and the GPU memory consumption. An overview of the main results is presented in Table 1.

Implementation Details. For Infinity, we grouped scales 10, 11, 12, and 13 into a single group, with token retention ratios of [20%, 10%, 5%, 1%], respectively. For HART, scales 12 and 13 were grouped, with token retention ratios of [30%, 20%]. The selection of grouped scales and their corresponding retention ratios was primarily guided by the distribution of $\mathrm{Update_Index}_k$. See the ablation study for a more detailed discussion. All models were implemented with FlashAttention [11, 12] acceleration, and our pruning method is fully compatible with it. We used the retained token indices to recover the original positional encodings, ensuring that pruning did not disturb the relative positional relationships among tokens. Additionally, we incorporated attention masks similar to those used during VAR model training to preserve causal dependencies between scales within the same group during decoding. All experiments were conducted on a single RTX 4090 GPU.

4.2. Comparison Experiments

Comparison on GenEval. We first evaluated the performance of our LazyVAR method on the GenEval benchmark to assess its capability in high-level semantic 1024×1024 text-to-image generation. Input prompts are rewritten using the Qwen2-VL-7B [65] model. The results are shown in Tab. 2. The baseline methods include diffusion-based and AR/MAR-based image generation models. Compared with other MAR/AR-based models, LazyVAR achieves significantly faster generation while maintaining superior image quality. Relative to the original Infinity model, our Infinity+LazyVAR achieves a 2.94× acceleration with even a slight performance improvement of 0.3%. For HART+LazyVAR, we observe a 1.67× acceleration accompanied by only a marginal performance decrease of 0.8%. For both VAR families, HART and Infinity, integrating LazyVAR consistently outperforms FastVAR in terms of both speed and quality. Moreover, our approach demonstrates lower memory consumption and fewer FLOPs than the previous FastVAR method. These results confirm the effectiveness and generalization capability of our method compared with existing approaches.

Comparison on MJHQ-30K. To further evaluate the perceptual quality of image generation, we conducted experiments on the MJHQ-30K benchmark. Table 1 presents the overall results, while Table 3 shows the results broken down by category. The results show that our LazyVAR maintains the image generation performance while achieving a high acceleration ratio. Interestingly, HART+FastVAR even achieves a lower FID [26] than the unpruned HART; how-

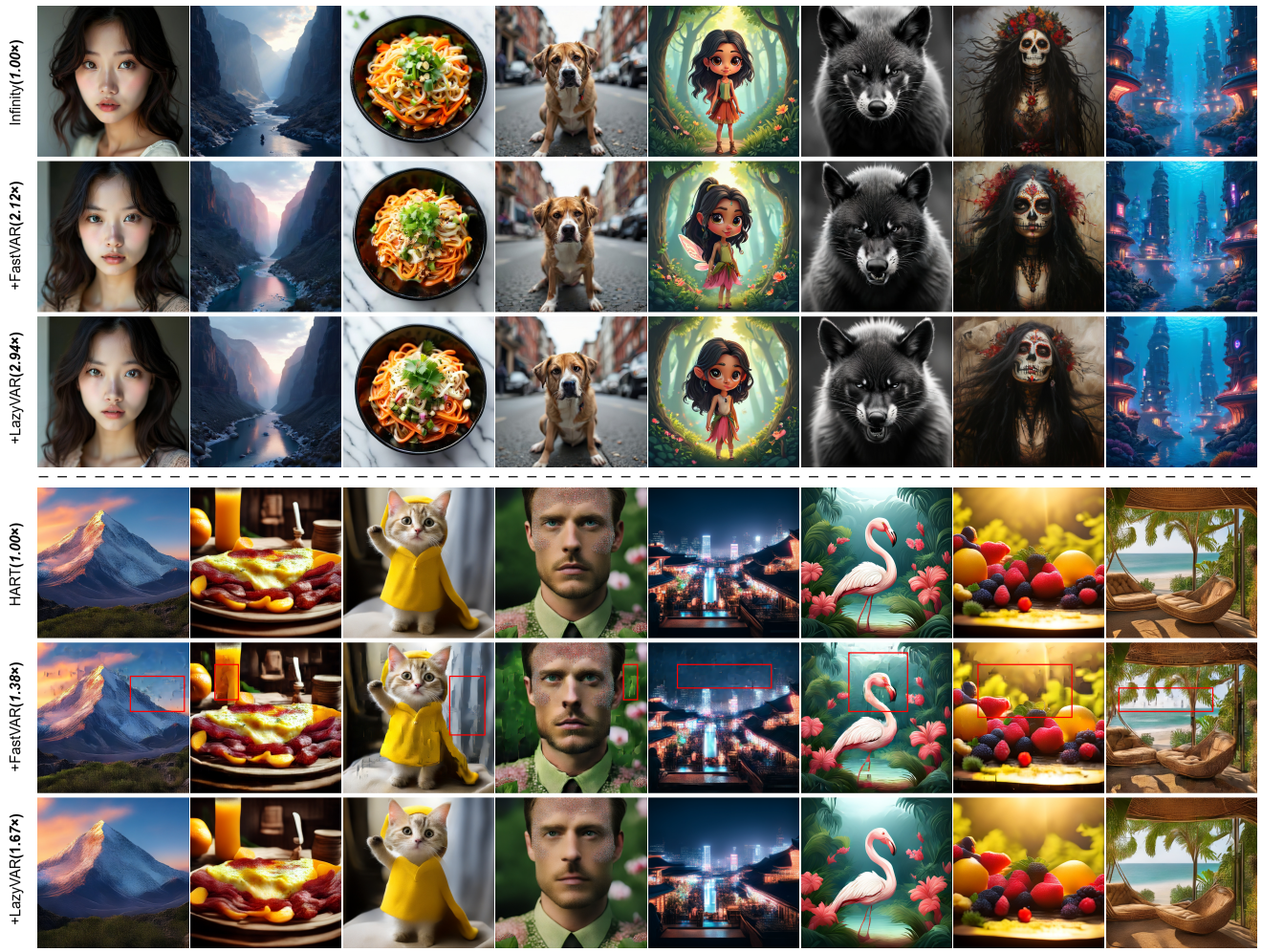


Figure 5. Qualitative comparison among the original models and FastVAR baseline and our LazyVAR on 1024×1024 image generation.

Table 4. Quantitative comparisons of FID* across all categories in the MJHQ-30K benchmark.**Average** is computed over the 10 categories.

Methods	Speed-up↑	animals	art	fashion	food	indoor	landscape	logo	people	plants	vehicles	Average
Infinity [22]	1.00×	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
+ FastVAR [21]	2.12×	11.19	13.23	12.36	12.14	13.45	10.64	13.54	14.34	13.12	11.14	12.52
+ LazyVAR	**2.94**×	**7.77**	**8.21**	**7.42**	**7.36**	**8.00**	**7.44**	**6.81**	**8.94**	**7.81**	**7.12**	**7.69**
HART [55]	1.00×	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
+ FastVAR [21]	1.38×	13.29	15.97	19.35	17.24	16.57	13.51	28.68	20.66	25.99	13.76	18.50
+ LazyVAR	**1.67**×	**10.50**	**13.05**	**11.40**	**12.38**	**12.56**	**12.79**	**13.03**	**13.64**	**13.80**	**12.32**	**12.55**

ever, we observed that it introduces visual artifacts in the background regions, as highlighted by the red boxes in Figure 5. In contrast, our HART+LazyVAR attains an FID comparable to that of the original HART and maintains visually indistinguishable image quality, while achieving a 1.67× acceleration. To further assess the visual consistency of generated images before and after pruning, we additionally report FID*, which measures the FID between images generated by the original and the pruned models. As shown in Table 4, LazyVAR consistently achieves lower FID* values than FastVAR across all generation categories. We also provide qualitative visualizations in Figure 5, which clearly demonstrate that images generated by LazyVAR preserve exceptionally high perceptual quality and accurate semantic alignment, confirming the effectiveness of LazyVAR.

Comparison on HPSv2.1. We further evaluate LazyVAR using the HPSv2.1 benchmark. As shown in Table 5, Infinity+LazyVAR achieves a 2.94× acceleration with only a 0.61 decrease in the metric. Moreover, LazyVAR surpasses FastVAR in both speed and generation quality.

Table 5. Comprehensive quantitative evaluation on HPSv2.1.

Model	Anime	Art	Paintings	Photo	Average
Infinity [22]	31.67	30.33	30.42	29.35	30.44
+FastVAR [21]	30.83	29.67	**29.73**	28.67	29.73
+LazyVAR	**31.04**	**29.72**	29.72	**28.85**	**29.83**
HART [55]	30.64	28.86	28.68	27.48	28.92
+FastVAR [21]	28.33	26.91	26.93	25.48	26.91
+LazyVAR	**29.14**	**27.20**	**26.94**	**26.19**	**27.36**

Table 6. Ablation on Pruning ratio. † indicates the default setting.

	Inference Efficiency		GenEval	MJHQ-30K		
Model	Time↓	Speed-up↑	Overall↑	FID↓	FID*↓	CLIP↑
Infinity [22]	1.38 s	1.00×	0.685	9.80	0.00	0.277
[20%, 10%]†	0.47 s	2.94×	0.686	9.83	1.46	0.277
[0%,0%]	0.41 s	3.37×	0.658	10.81	2.67	0.277
[40%, 20%]	0.51 s	2.71×	0.686	9.87	1.65	0.276
[60%, 30%]	0.56 s	2.46×	0.685	9.82	1.52	0.277
FastVAR* [21]	0.55 s	2.51×	0.662	10.15	2.38	0.278

4.3. Ablation Studies

Different pruning ratios. Consistent with prior work, pruning is applied only at the larger scales. We ablate the proportion of tokens retained at scales 9 and 10 of the Infinity model. As shown in the table, removing all tokens at these two steps causes a severe degradation in generation quality, whereas retaining a moderate fraction yields a favorable trade-off between quality and efficiency. Our default retention ratios are $[20\%, 10\%]$. Increasing the retention ratios increases latency but does not always lead to improved generation performance. This observation is consistent with the analysis above because many tokens at the larger scales receive only very small updates; excluding them from update does not harm perceptual quality. For a more stringent comparison, we also increased the pruning ratios of FastVAR to further reduce latency, denoted as FastVAR*. Due to the lack of parallelism, FastVAR* could not achieve the same acceleration ratio as LazyVAR and suffered a marked decline in generation quality.

Different grouping strategy. We ablate the effect of grouping parallel scales for decoding. The grouping $[9, 10, 11, 12, 13]$ denotes treating these scales as a single group. Grouping more steps increases throughput but may degrade image quality. Compared with decoding without grouping, LazyVAR attains a 1.45× speedup while preserving nearly identical generation quality.

Effectiveness of the pruning criterion. Replacing our pruning criterion with the PTS used in FastVAR results in a substantial deterioration of image quality. Omitting the use of the cumulative retention tokens derived from Equation (10), which ensures that the same token position is not updated across different scales within the same group, results in a slight performance drop.

Table 7. Ablation on grouping strategy.† means the default setting.

	Inference Efficiency		GenEval	MJHQ-30K		
Model	Time↓	Speed-up↑	Overall↑	FID↓	FID*↓	CLIP↑
Infinity [22]	1.38 s	1.00×	0.685	9.80	0.00	0.277
[10,11,12,13]†	0.47 s	2.94×	0.686	9.83	1.46	0.277
w/o PGD	0.68 s	2.03×	0.690	9.74	1.48	0.277
[9,10,11,12,13]	0.42 s	3.29×	0.678	9.83	2.21	0.276
[10,11],[12,13]	0.52 s	2.65×	0.680	9.74	1.52	0.277

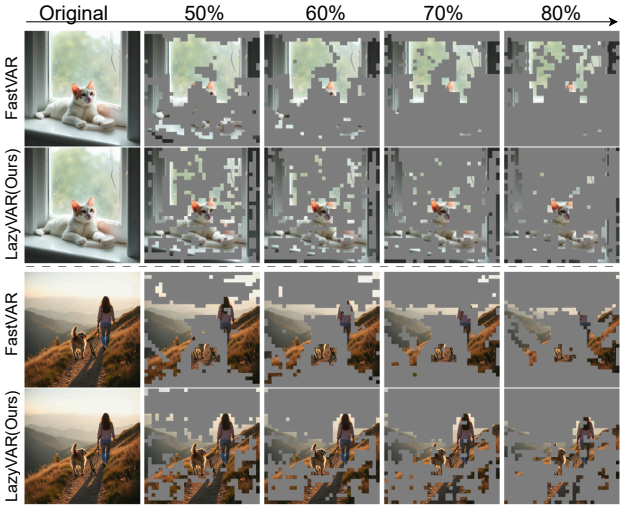


Figure 6. Visualization of pruned tokens under different ratios.

Table 8. Ablation on Pruning criterion.† means the default setting.

	Inference Efficiency		GenEval	MJHQ-30K		
Model	Time↓	Speed-up↑	Overall↑	FID↓	FID*↓	CLIP↑
Infinity [22]	1.38 s	1.00×	0.685	9.80	0.00	0.277
w UIGTP†	0.47 s	2.94×	0.686	9.83	1.46	0.277
w/o Cum	0.47 s	2.94×	0.681	9.88	1.54	0.277
w PTS	0.48 s	2.88×	0.665	10.12	2.29	0.278

5. Conclusion

In summary, this work presents **LazyVAR**, a method to accelerate text-to-image synthesis with VAR models. We show that the cosine similarity of aggregated latent features between adjacent scales exhibits a characteristic long-tailed distribution, with the vast majority of tokens displaying high similarity at larger scales. Based on this observation, we propose a scale-wise update index guided pruning strategy to address the excessive number of tokens processed at large scales. In addition, we introduce a parallel group decoding scheme to overcome the inherent limitation that different steps cannot be processed in parallel. Experimental results confirm that LazyVAR achieves substantial acceleration while largely preserving generation quality, demonstrating its effectiveness for VAR-based text-to-image models acceleration.

Acknowledgement

This work was supported by Natural Science Foundation of China under Grant 62271465, Suzhou Basic Research Program under Grant SYG202338, and Open Fund Project of Guangdong Academy of Medical Sciences, China (No. YKY-KF202206).

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[2] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 2

[3] Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *CVPR*, pages 4599–4603, 2023. 2

[4] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *CVPR*, pages 11315–11325, 2022. 2

[5] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023. 2

[6] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. In *ECCV*, pages 74–91. Springer, 2024. 5, 6

[7] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, pages 1691–1703. PMLR, 2020. 2

[8] Zigeng Chen, Xinyin Ma, Gongfan Fang, and Xinchao Wang. Collaborative decoding makes visual auto-regressive modeling efficient. *arXiv preprint arXiv:2411.17787*, 2024. 2

[9] Zhuokun Chen, Jugang Fan, Zhuowei Yu, Bohan Zhuang, and Mingkui Tan. Frequency-aware autoregressive modeling for efficient high-resolution image synthesis. In *ICCV*, pages 17140–17149, 2025. 2

[10] Quan Dao, Xiaoxiao He, Ligong Han, Ngan Hoai Nguyen, Amin Heyrani Nobar, Faez Ahmed, Han Zhang, Viet Anh Nguyen, and Dimitris Metaxas. Discrete noise inversion for next-scale autoregressive text-based image editing. *arXiv preprint arXiv:2509.01984*, 2025. 2

[11] Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *ICLR*, 2024. 6

[12] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flash Attention: Fast and memory-efficient exact attention with IO-awareness. In *NeurIPS*, 2022. 6

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019. 2

[14] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, pages 8780–8794, 2021. 1

[15] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, pages 12873–12883, 2021. 1, 2

[16] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024. 5, 6

[17] Qichen Fu, Minsik Cho, Thomas Merth, Sachin Mehta, Mohammad Rastegari, and Mahyar Najibi. LazyLLM: Dynamic token pruning for efficient long context llm inference. *arXiv preprint arXiv:2407.14057*, 2024. 3

[18] Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. Model tells you what to discard: Adaptive KV cache compression for LLMs. *arXiv preprint arXiv:2310.01801*, 2023. 2

[19] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *NeurIPS*, 2024. 5

[20] Hang Guo, Yawei Li, Tao Dai, Shu-Tao Xia, and Luca Benini. Intlora: Integral low-rank adaptation of quantized diffusion models. *arXiv preprint arXiv:2410.21759*, 2024. 2

[21] Hang Guo, Yawei Li, Taolin Zhang, Jiangshan Wang, Tao Dai, Shu-Tao Xia, and Luca Benini. Fastvar: Linear visual autoregressive modeling via cached token pruning. *arXiv preprint arXiv:2503.23367*, 2025. 2, 3, 5, 6, 7, 8

[22] Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. *arXiv preprint arXiv:2412.04431*, 2024. 2, 3, 5, 6, 7, 8

[23] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022. 2

[24] Yefei He, Feng Chen, Yuanyu He, Shaoxuan He, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. ZIPAR: Accelerating autoregressive image generation through spatial locality. *arXiv preprint arXiv:2412.04062*, 2024. 3

[25] Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020. 2

[26] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 6

[27] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, pages 6840–6851, 2020. 1

[28] Doohyuk Jang, Sihwan Park, June Yong Yang, Yeonsung Jung, Jihun Yun, Souvik Kundu, Sung-Yub Kim, and Eunho Yang. Lantern: Accelerating visual autoregressive models with relaxed speculative decoding. *arXiv preprint arXiv:2410.03355*, 2024. 3

[29] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 2

[30] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *ICML*, pages 19274–19286. PMLR, 2023. 3

[31] Daiqing Li, Aleks Kamko, Ehsan Akhgari, Ali Sabet, Linmiao Xu, and Suhail Doshi. Playground v2.5: Three insights towards enhancing aesthetic quality in text-to-image generation. *arXiv preprint arXiv:2402.17245*, 2024. 5

[32] Jiajun Li, Yue Ma, Xinyu Zhang, Qingyan Wei, Songhua Liu, and Linfeng Zhang. Skipvar: Accelerating visual autoregressive modeling via adaptive frequency-aware skipping. *arXiv preprint arXiv:2506.08908*, 2025. 2

[33] Kunjun Li, Zigeng Chen, Cheng-Yen Yang, and Jenq-Neng Hwang. Memory-efficient visual autoregressive modeling with scale-aware kv cache compression. *arXiv preprint arXiv:2505.19602*, 2025. 2

[34] Senmao Li, Taihang Hu, Fahad Shahbaz Khan, Linxuan Li, Shiqi Yang, Yaxing Wang, Ming-Ming Cheng, and Jian Yang. Faster Diffusion: Rethinking the role of unet encoder in diffusion models. In *NeurIPS*, 2023. 2

[35] Tianhong Li, Huiwen Chang, Shlok Mishra, Han Zhang, Dina Katabi, and Dilip Krishnan. Mage: Masked generative encoder to unify representation learning and image synthesis. In *CVPR*, pages 2142–2152, 2023. 2

[36] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *arXiv preprint arXiv:2406.11838*, 2024. 2

[37] Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-Diffusion: Quantizing diffusion models. In *ICCV*, pages 17535–17545, 2023. 2

[38] Xuejie Liu, Anji Liu, Guy Van den Broeck, and Yitao Liang. Plug-and-play context feature reuse for efficient masked generation. *arXiv preprint arXiv:2505.19089*, 2025. 3

[39] Yi Liu, Shengqian Li, Zuzeng Lin, Feng Wang, and Si Liu. Cyclevar: Repurposing autoregressive model for unsupervised one-step image translation. In *ICCV*, pages 15194–15204, 2025. 2

[40] Xinyin Ma, Gongfan Fang, Michael Bi Mi, and Xinchao Wang. Learning-to-Cache: Accelerating diffusion transformer via layer caching. *arXiv preprint arXiv:2406.01733*, 2024. 2

[41] Xinyin Ma, Gongfan Fang, and Xinchao Wang. DeepCache: Accelerating diffusion models for free. In *CVPR*, pages 15762–15772, 2024. 2

[42] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *CVPR*, pages 14297–14306, 2023. 2

[43] Ziqi Pang, Tianyuan Zhang, Fujun Luan, Yunze Man, Hao Tan, Kai Zhang, William T Freeman, and Yu-Xiong Wang. Randar: Decoder-only autoregressive visual generation in random orders. In *CVPR*, pages 45–55, 2025. 2

[44] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 1

[45] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *NeurIPS*, 2019. 2

[46] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 1

[47] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 2

[48] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 2

[49] Andrea Santilli, Silvio Severino, Emilian Postolache, Valentino Maiorca, Michele Mancusi, Riccardo Marin, and Emanuele Rodolà. Accelerating transformer inference for translation via parallel decoding. *arXiv preprint arXiv:2305.10427*, 2023. 3

[50] Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *CVPR*, pages 1972–1981, 2023. 2

[51] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, pages 2256–2265. pmlr, 2015. 1

[52] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2

[53] Charles Spearman. The proof and measurement of association between two things. *The American journal of psychology*, 100(3/4):441–471, 1987. 4

[54] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024. 1, 5

[55] Haotian Tang, Yecheng Wu, Shang Yang, Enze Xie, Junsong Chen, Junyu Chen, Zhuoyang Zhang, Han Cai, Yao Lu, and Song Han. HART: Efficient visual generation with hybrid autoregressive transformer. *arXiv preprint arXiv:2410.10812*, 2024. 2, 3, 5, 6, 7, 8

[56] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024. 2

[57] Yao Teng, Han Shi, Xian Liu, Xuefei Ning, Guohao Dai, Yu Wang, Zhenguo Li, and Xihui Liu. Accelerating auto-regressive text-to-image generation with training-free spec-

ulative jacobi decoding. *arXiv preprint arXiv:2410.01699*, 2024. 3

[58] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *NeurIPS*, 2024. 2

[59] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2

[60] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *NeurIPS*, 2016. 2

[61] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *NeurIPS*, 2017. 1, 2

[62] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[63] Hongjie Wang, Difan Liu, Yan Kang, Yijun Li, Zhe Lin, Niraj K Jha, and Yuchen Liu. Attention-driven training-free efficiency enhancement of diffusion models. In *CVPR*, pages 16080–16089, 2024. 2

[64] Jiahuan Wang, Yuxin Chen, Jun Yu, Guangming Lu, and Wenjie Pei. Editinfinity: Image editing with binary-quantized generative models. *arXiv preprint arXiv:2510.20217*, 2025. 2

[65] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 6

[66] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. EMU3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024. 2

[67] Yufei Wang, Lanqing Guo, Zhihao Li, Jiaxing Huang, Pichao Wang, Bihan Wen, and Jian Wang. Training-free text-guided image editing with visual autoregressive model. In *ICCV*, pages 17577–17586, 2025. 2

[68] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023. 5

[69] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023. 3

[70] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024. 5

[71] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *CVPR*, pages 9653–9663, 2022. 2

[72] Feihong Yan, Qingyan Wei, Jiayi Tang, Jiajun Li, Yulin Wang, Xuming Hu, Huiqi Li, and Linfeng Zhang. Lazymar: Accelerating masked autoregressive models via feature caching. In *ICCV*, pages 15552–15561, 2025. 3

[73] Hu Yu, Hao Luo, Hangjie Yuan, Yu Rong, and Feng Zhao. Frequency autoregressive image generation with continuous tokens. *arXiv preprint arXiv:2503.05305*, 2025. 2

[74] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Randomized autoregressive visual generation. In *ICCV*, pages 18431–18441, 2025. 2

[75] Evelyn Zhang, Bang Xiao, Jiayi Tang, Qianli Ma, Chang Zou, Xuefei Ning, Xuming Hu, and Linfeng Zhang. Token pruning for caching better: 9 times acceleration on stable diffusion for free. *arXiv preprint arXiv:2501.00375*, 2024. 2

[76] Jinhua Zhang, Wei Long, Minghao Han, Weiyi You, and Shuhang Gu. Mvar: Visual autoregressive modeling with scale and spatial markovian conditioning. *arXiv preprint arXiv:2505.12742*, 2025. 2

[77] Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching. *arXiv preprint arXiv:2410.05317*, 2024. 2