Frequency-Aware Autoregressive Modeling for Efficient High-Resolution Image Synthesis

Zhuokun Chen[1,2*] Jugang Fan[1,2] Zhuowei Yu[3] Bohan Zhuang[4†] Mingkui Tan[1,2†]
[1] South China University of Technology [2] Pazhou Lab [3] University of California, Davis [4] Zhejiang University

Abstract

Visual autoregressive modeling, based on the next-scale prediction paradigm, exhibits notable advantages in image quality and model scalability over traditional autoregressive and diffusion models. It generates images by progressively refining resolution across multiple stages. However, the computational overhead in high-resolution stages remains a critical challenge due to the substantial number of tokens involved. In this paper, we introduce SparseVAR, a plug-and-play acceleration framework for next-scale prediction that dynamically excludes low-frequency tokens during inference without requiring additional training. Our approach is motivated by the observation that tokens in low-frequency regions have a negligible impact on image quality in high-resolution stages and exhibit strong similarity with neighboring tokens. Additionally, we observe that different blocks in the next-scale prediction model focus on distinct regions, with some concentrating on high-frequency areas. SparseVAR leverages these insights by employing lightweight MSE-based metrics to identify low-frequency tokens while preserving the fidelity of excluded regions through a small set of uniformly sampled anchor tokens. By significantly reducing the computational cost while maintaining high image generation quality, SparseVAR achieves notable acceleration in both HART and Infinity. Specifically, SparseVAR achieves up to a 2× speedup with minimal quality degradation in Infinity-2B. Code is available at https://github.com/Caesarhhh/SparseVAR.

1. Introduction

Text-to-image generation has seen widespread application across a range of fields, from creative industries to practical domains like virtual reality and content creation [6, 17, 22, 24, 25, 31, 35]. Among the various approaches, autoregressive models [7, 14, 18–20, 27, 32, 37] stand out by utilizing a pre-trained tokenizer to quantize continuous image features into a sequence of discrete features by referring to a codebook. This allows the model to generate images by predicting the tokens in a sequence, achieving impressive generalization and scalability. Building on this, next-scale prediction [10, 28, 29] further accelerates autoregressive inference by progressively increasing image resolution and predicting the token maps of each resolution stage. This approach generates images in multiple stages, with each stage progressively increasing the resolution. By predicting an entire resolution at each stage, the number of iterations required for high-resolution image generation is significantly reduced. However, during the high-resolution stages, next-scale prediction typically requires the generation of thousands of tokens per resolution, leading to substantial computational overhead, presenting a major challenge for scaling autoregressive models in high-resolution image synthesis.

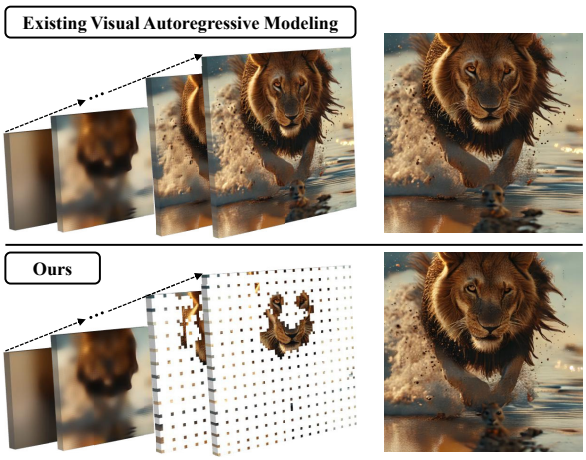

Figure 1. Existing visual autoregressive models allocate uniform computational resources across all regions of a high-resolution image. However, the large number of tokens processed in parallel during high-resolution stages leads to substantial computational overhead. To address this, our method decomposes the target image into high- and low-frequency components, effectively reducing the computational cost in high-resolution stages, thereby lowering inference latency while preserving image generation quality.

To reduce the computational burden during high-resolution stages, a natural approach is to decrease the num-

*Email: caesard216@gmail.com
†Corresponding author. Email: bohan.zhuang@gmail.com , mingkuitan@scut.edu.cn

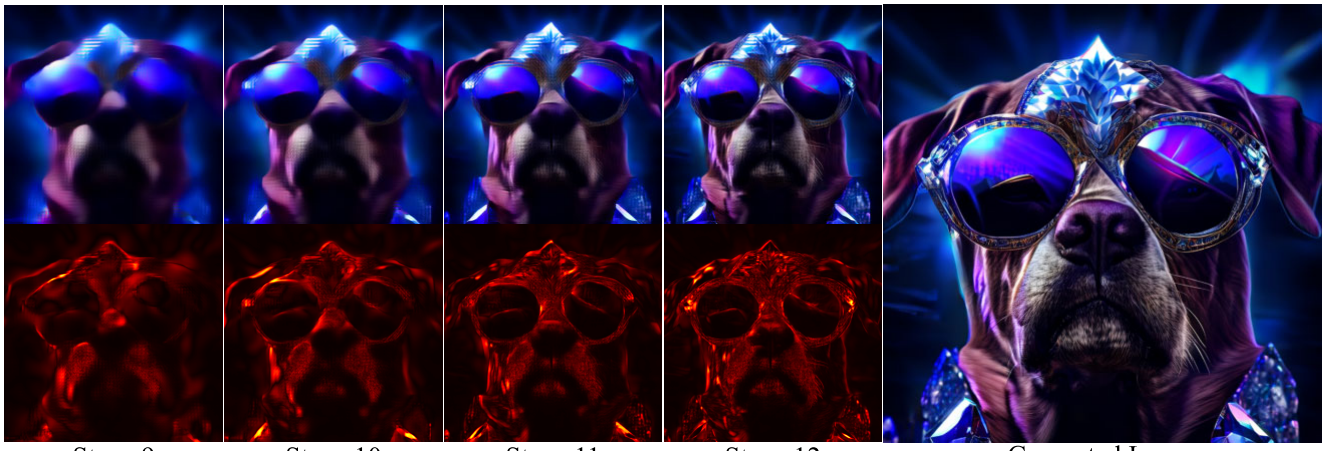

Figure 2. **High-resolution stages have minimal impact on low-frequency regions.** We visualize the images generated by the last five higher-resolution stages of HART-0.7B (top), along with the ℓ_1 difference between each stage and its previous stage (bottom). The redder areas indicate where the ℓ_1 difference is larger, and these areas are mostly concentrated in the high-frequency regions of the foreground. In contrast, the ℓ_1 change in low-frequency regions, such as the background, is minimal, highlighting that high-resolution stages predominantly focus on high-frequency regions.

ber of tokens involved in the computation. Previous works have extensively explored methods to reduce the number of tokens in vision transformers and multimodal large language models [1, 3, 5, 26], which can be broadly categorized into two strategies: merging and selection. Token merging [3, 16, 26] exploit the inherent similarity across visual tokens, using similarity matching or clustering to combine similar tokens. However, in high-resolution image generation, the large number of tokens involved makes techniques like clustering and similarity matching computationally prohibitive. Token selection [1, 5, 12] relies on the redundancy of token attention scores, removing tokens with low attention scores in earlier layers to reduce the token count in subsequent layers, or applying sparse attention operators to accelerate computation. However, as shown in previous work [11, 29], autoregressive image generation models exhibit strong local dependencies in token attention scores, where nearly all tokens assign high attention scores to their neighboring tokens and low scores to distant tokens. This consistent pattern across tokens makes it difficult to distinguish redundant tokens based solely on attention scores. Detailed visualizations of the next-scale prediction model's attention map are provided in the appendix.

Our analysis begins by exploring the redundancy of token maps across different stages. As shown in Figure 2, we visualize the ℓ_1 difference between images generated with and without the residuals from the final several stages of HART-0.7B [28]. The residuals are concentrated in high-frequency regions, while the impact of the final stage's residuals on low-frequency regions is negligible. This indicates substantial redundancy in token inference during high-resolution stages. Next, as shown in Figure 3, we visualize the MSE changes in the features before and after inference across different blocks of the HART-0.7B. Our observations indicate that different blocks attend to distinct regions, with certain blocks exhibiting pronounced changes in high-frequency regions.

Based on the above observations, we propose SparseVAR, a plug-and-play method designed to accelerate any next-scale prediction model without the need for additional training. Starting from a relatively high-resolution stage, SparseVAR dynamically identifies low-frequency tokens using a lightweight metric based on the MSE changes observed across features at specific blocks focusing on high-frequency regions, eliminating the need for computationally expensive similarity matching. Tokens identified as low-frequency are skipped in subsequent inference stages. Moreover, SparseVAR opts to retain a small number of anchor tokens, which serve as proxies to represent the low-frequency regions, to effectively preserve the generation quality while incurring minimal additional computation.

We evaluate SparseVAR on leading high-resolution next-scale prediction methods. The results demonstrate that our method significantly accelerates image generation with virtually no loss in quality. For instance, on the GenEval dataset, SparseVAR improves the inference speed of Infinity by an average of nearly $2\times$ with a minimal quality degradation in the generated images.

Overall, our contributions are as follows:

- We offer new insights into next-scale prediction models: (1) A significant amount of redundant tokens exist during inference at high-resolution stages, (2) Different blocks focus on distinct regions.
- We introduce SparseVAR, a simple yet effective method for accelerating next-scale prediction models. Sparse-

VAR dynamically identifies low-frequency tokens using a lightweight metric, enabling early exclusion of low-frequency tokens during high-resolution stages, thus significantly reducing computational overhead in low-frequency regions. Moreover, SparseVAR preserves the generation quality of low-frequency regions by retaining specific anchor tokens.

2. Related Work

Next-scale prediction. Next-scale prediction [10, 28, 29], first introduced by VAR [29], demonstrates the potential of the autoregressive paradigm in image generation, rivaling diffusion transformers [2, 4, 21]. Traditional autoregressive (AR) models [8, 15, 23, 36] flatten 2D images into 1D sequences of patch-level tokens. However, the spatial locality inherent in images leads to strong correlations among neighboring patches, which conflicts with the unidirectional dependency assumption in AR modeling and limits both scalability and generalization. VAR [29] addresses this limitation by employing a multi-scale VQ-VAE [30] to represent images as multi-scale token maps. In this framework, each scale's token map is treated as an autoregressive unit, progressively predicting higher-resolution token maps at each step. While effective, the discrete tokenizer [30] used in VAR struggles to recover fine-grained image details, imposing an upper bound on generation quality. HART [28] mitigates this issue by introducing a continuous-discrete hybrid tokenizer, significantly improving generation quality at higher resolutions. Inspired by binary vector quantization [38], Infinity [10] further expands the tokenizer vocabulary and adopts bitwise token prediction, enabling more detailed reconstructions. Despite these advances, these models face challenges related to high computational redundancy, particularly during the last few high-resolution stages of generation.

Token reduction. Reducing the number of input tokens is a common strategy to enhance computational efficiency. Existing approaches primarily employ token selection [1, 5, 12] or token merging [3, 16, 26]. FastV [5] ranks tokens based on their attention scores up to the K-th layer and prunes the bottom $R\%$, retaining the remaining tokens for subsequent processing. HiRED [1] addresses high-resolution image inputs by dynamically allocating token budgets per sub-image using shallow-layer attention and selecting the top N patches per sub-image based on deeper-layer [CLS] attention. Similarly, ZipVL [12] employs an adaptive ratio assignment scheme to discard less critical tokens, thereby compressing the KV cache and accelerating the attention operation. However, token selection methods are unsuitable for generative models due to the high interdependence of tokens. Alternatively, token merging approaches reduce redundancy by combining similar tokens. ToMe [3] divides tokens into two groups, calculates inter-group similarity, and merges the top N pairs of most similar tokens. VTM [16] introduces a learnable token merging technique for long-form video inputs, considering both token similarity and saliency. LLava-PruMerge [26] integrates selection and merging by initially selecting visual tokens based on [CLS] attention scores, followed by merging using k-nearest neighbors. In the context of high-resolution image generation, the sheer volume of tokens significantly amplifies the computational cost of clustering and similarity matching, rendering such techniques infeasible for practical applications.

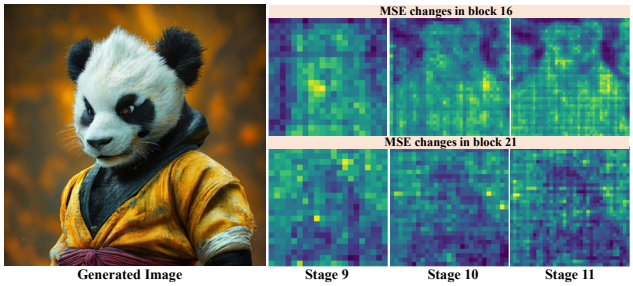

Figure 3. **Different blocks in next-scale prediction models tend to focus on distinct regions.** We visualize the MSE changes before and after feature inference at the 16th and 21st blocks during stages 10-12 of HART-0.7B. It is clear that different blocks exhibit distinct regional focus tendencies.

3. Empircal Insights

In this section, we provide visualizations and a in-depth analysis of next-scale prediction models, revealing two key properties that offer critical insights for our method.

Observation 1: The residuals generated at high-resolution stages have minimal impact on low-frequency regions. Existing VAR models predict logits $\tilde{\mathbf{p}}_k$ at each stage k, which are then mapped to residual feature maps Δr_k via the pre-trained codebook. To investigate the influence of these predictions on the final image, especially at high-resolution stages, we visualize the ℓ_1 changes between decoded images from two adjacent stages. As shown in Figure 2, with increasing stages, the residuals concentrate on high-frequency regions, while their effect on the majority of low-frequency regions is minimal. This indicates significant redundancy in the high-resolution stages. Inspired by this, *we propose early exclusion of low-frequency tokens at high-resolution stages to reduce this redundancy.*

Observation 2: Different blocks in next-scale prediction models tend to focus on distinct regions. To investigate the regional differences in focus across blocks, we visualize the MSE changes in features before and after inference at various blocks. As shown in Figure 3, the regions attended differ significantly across blocks. Specifically, the block 16 focuses more on high-frequency regions like contours, while block 21 emphasizes low-frequency regions such as background. Based on this observation, *we propose dynamically*

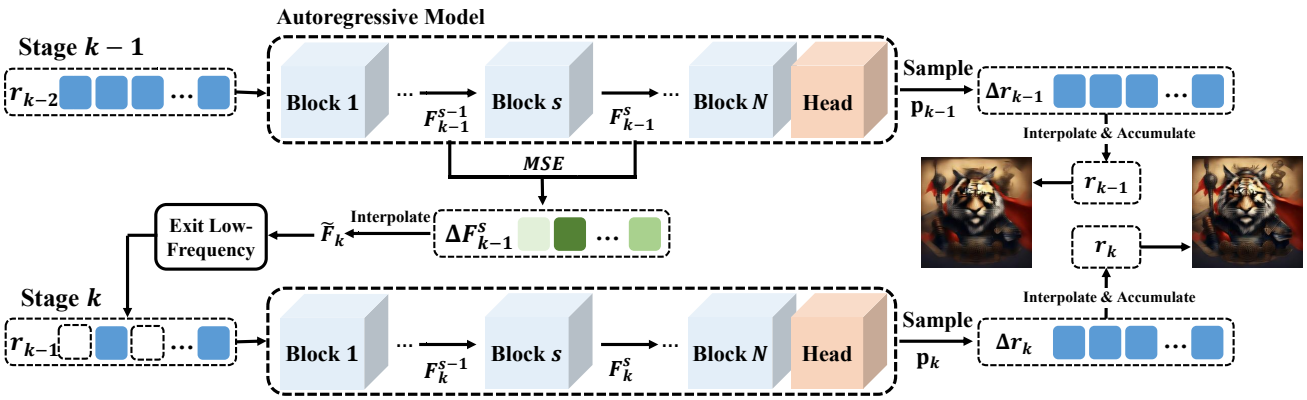

Figure 4. Overview of dynamic exclusion in SparseVAR. SparseVAR dynamically identifies and excludes low-frequency tokens starting from higher-resolution stages by analyzing MSE changes in features before and after inference in specific blocks, which significantly reduces computational overhead while maintaining generation quality of high-resolution regions.

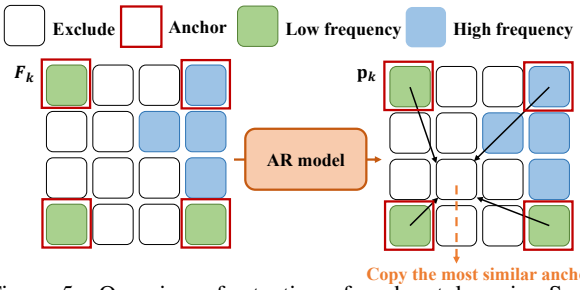

Figure 5. Overview of retention of anchor tokens in SparseVAR ($\alpha = 3$). SparseVAR retains a small number of anchor tokens to represent low-frequency regions, enabling early-excluded tokens to copy predictions from the most similar anchors.

distinguishing high- and low-frequency regions of the generated image using the MSE changes in features before and after inference in specific blocks.

4. Methodology

Inspired by the above observations, we propose SparseVAR, a simple yet effective method for accelerating next-scale prediction models. SparseVAR comprises two key components: early exclusion of low-frequency tokens and the retention of anchor tokens. As shown in Figure 4, Figure 5 and Algorithm 1, SparseVAR identifies low-frequency tokens by analyzing the MSE changes at specific blocks, enabling their early exclusion to reduce computational cost. Simultaneously, SparseVAR retains a set of anchor tokens to ensure the preservation of generation quality in low-frequency regions. The acceleration provided by SparseVAR is plug-and-play, making it compatible with any next-scale prediction model without the need for additional training.

4.1. Preliminary

Inference of the next-scale prediction. Consider a next-scale prediction model comprising N blocks, the VAR framework employs a hierarchical generation process across K progressive resolution scales. At each scale $k \in \{1, ..., K\}$, the model parallelly predicts logits $\mathbf{p}_k$ for all $h_k \times w_k$ tokens in the current resolution scale. Subsequently, the residual feature map Δr_k is generated by retrieving features for each token from the pretrained codebook based on the predicted logit map $\mathbf{p}_k$. The residual feature Δr_i from all previous stages ($i \leq k$) are interpolated and accumulated to form r_k, which serves as the input for stage $k+1$. Finally, r_K is used to generate the final image through a VAE decoder.

4.2. Dynamic Exclusion of Low-Frequency Tokens

Exclusion in high-resolution stages. As illustrated in **Observation 1**, we propose to exclude low-frequency tokens during inference to reduce computational overhead. Considering that earlier stages of the next-scale prediction model have lower computational overhead and mainly capture low-frequency information (hence exhibiting limited redundancy), we only start applying early-exit from the P-th stage onward.

Dynamic high-low frequency identification. Since the proportion of low-frequency regions varies across images, it is essential to design a lightweight method to dynamically identify regions that are low-frequency and should be excluded from the computation. Inspired by **Observation 2**, we directly leverage the MSE variations of features within a specific block, to effectively distinguish high- and low-frequency regions. As illustrated in Figure 4, let $\mathbf{F}^s_k \in \mathbb{R}^{h_k \times w_k \times C}$ represents the output feature map of selected block s at stage $k (k \geq P)$, where C is the number of channels. The MSE change map $\Delta\mathbf{F}^s_{k-1} \in \mathbb{R}^{h_{k-1} \times w_{k-1}}$ in previous stage is defined as:

$$\Delta\mathbf{F}^s_{k-1}(i,j) = \frac{1}{C}\sum_{c=1}^{C}\left(\mathbf{F}^s_{k-1}(i,j,c) - \mathbf{F}^{s-1}_{k-1}(i,j,c)\right)^2, \tag{1}$$

where i and j index the spatial dimensions of the feature map. $\Delta\mathbf{F}^s_{k-1}$ is interpolated to match the resolution of the stage k, resulting in $\tilde{\mathbf{F}}_k \in \mathbb{R}^{h_k \times w_k}$.

Let $\mathcal{M}^{\text{low}}_k \subseteq \{1, \ldots, h_k\} \times \{1, \ldots, w_k\}$ denote the set of low-frequency token indices at stage k. Tokens in stage k are classified as low-frequency and added to the exclusion set $\mathcal{M}^{\text{low}}_k$ if their corresponding values in $\tilde{\mathbf{F}}_k$ are below a threshold $\tau \cdot \max(\tilde{\mathbf{F}}_k)$, where $\tau \in [0, 1]$ is a hyperparameter controlling the sparsity level. Formally, we define:

$$\mathcal{M}^{\text{low}}_k = \{(i, j) \mid \tilde{\mathbf{F}}_k(i, j) < \tau \cdot \max(\tilde{\mathbf{F}}_k)\}. \tag{2}$$

Regions identified as low-frequency are excluded from computation in stage k and all subsequent stages, ensuring that they do not participate in any further computations.

4.3. Retention of Anchor Tokens

To ensure the generation quality of low-frequency regions, we propose retaining a set of anchor tokens that encapsulate the essential information of excluded low-frequency regions by leveraging the high similarity of tokens in neighboring low-frequency areas. Specifically, we uniformly select the top-left corner of every $\alpha \times \alpha$ grid as the anchor token, ensuring efficient representation while preserving essential structural information. As illustrated in Figure 5, when $\alpha = 3$, anchor tokens are selected from the top-left corner of every 3×3 grid.

At stage k ($k \geq P$), the output logits map $\mathbf{p}_{k-1}$ from the previous stage is utilized to assess the similarity of low-frequency regions. Specifically, $\mathbf{p}_{k-1}$ is interpolated to match the resolution of stage k, resulting in $\tilde{\mathbf{p}}_k$. The logits of anchor tokens, denoted as $\mathbf{a}_k$, are a subset of $\tilde{\mathbf{p}}_k$. The cosine similarity between $\tilde{\mathbf{p}}_k$ and $\mathbf{a}_k$ is computed as:

$$\text{Sim}(\tilde{\mathbf{p}}_k, \mathbf{a}_k) = \frac{\tilde{\mathbf{p}}_k^\top \mathbf{a}_k}{\|\tilde{\mathbf{p}}_k\| \|\mathbf{a}_k\|}. \tag{3}$$

After inference in stage k, an excluded token is assigned the logits of its most similar anchor token if the similarity exceeds a predefined threshold β; otherwise, the feature map Δr_k in its location is set to zero.

Complexity analysis. *For inference complexity, SparseVAR reduces the computational cost by excluding low-frequency tokens starting from stage P.* Since the computational cost of higher-resolution stages dominates the overall inference complexity in next-scale prediction models, we primarily analyze these stages.

For the k-th stage, the computational cost of the original model is $O\left(h_k^2 \cdot w_k^2\right)$. SparseVAR dynamically identifies the low-frequency regions, and suppose a proportion s_k of low-frequency tokens is excluded from computation. Additionally, anchor tokens are uniformly sampled from the feature map at each stage. The number of anchor tokens is proportional to $\frac{1}{\alpha^2}$, where α is the sampling size. The reduced computational cost at stage k, considering both the exclusion of low-frequency tokens and the inclusion of anchor tokens, becomes:

$$O\left((1 - s_k + \frac{1}{\alpha^2})^2 \cdot h_k^2 \cdot w_k^2\right).$$

Algorithm 1 Inference Procedure for Stage k in SparseVAR

Input: Input feature map r_{k-1}, selected block index s for MSE computation, threshold τ, anchor grid size α, similarity threshold β, $[\mathbf{p}_{k-1}, \Delta\mathbf{F}^s_{k-1}]$ (for $k \geq P$)
Output: Logits $\mathbf{p}_k$, MSE change map $\Delta\mathbf{F}^s_k$(for $k \geq P - 1$)

if $k < P$ **then**
  Directly inference and obtain logits $\mathbf{p}_k$
  if $k = P - 1$ **then**
    Compute MSE change map $\Delta\mathbf{F}^s_k$ using Eq. (1)
else
  // Exclude low-frequency tokens dynamically
  Interpolate $\Delta\mathbf{F}^s_{k-1}$ to resolution $\tilde{\mathbf{F}}^s_k$
  Identify low-frequency tokens $\mathcal{M}^{\text{low}}_k$ using Eq. (2)
  Exclude tokens in $\mathcal{M}^{\text{low}}_k$ except anchor tokens
  Inference and obtain logits $\mathbf{p}_k$ for remaining tokens
  // Copy logits from anchor token
  Interpolate logits $\mathbf{p}_{k-1}$ to match resolution the k-th stage and obtain $\tilde{\mathbf{p}}_k$
  for each token $(i, j) \in \mathcal{M}^{\text{low}}_k$ **do**
    Compute similarity with anchors using Eq. (3)
    if maximum similarity $\geq \beta$ **then**
      Assign logits of the most similar anchor
  Compute MSE change map $\Delta\mathbf{F}^s_k$ for the current stage using Eq. (1)
return $\mathbf{p}_k$, $\Delta\mathbf{F}^s_k$(for $k \geq P - 1$)

5. Experiments

Implementation details. We conduct experiments on the Infinity-2B [10] and HART-0.7B [28], both are 1024×1024 high-resolution autoregressive generation models based on next-scale prediction. We compare the performance of accelerated image generation on the GenEval [9], DPG-Bench [13], ImageReward [34], and HPSV2.1 [33] datasets with $\beta = 0.9$ and $P = 10$. Since the sparsity varies dynamically across images, we present the average per-image inference performance across these datasets. All inference latency is measured on an NVIDIA 3090 GPU.

Main results. To evaluate the acceleration performance of SparseVAR, we conduct experiments on the GenEval and DPG-Bench. As shown in Table 1 and Table 2, the results demonstrate that SparseVAR significantly improves inference speed with minimal impact on image generation quality. For instance, when $\tau = 0.7$, SparseVAR reduces the inference latency of Infinity by 51%, with the overall score

Table 1. Quantitative evaluation on GenEval. This table presents a comprehensive quantitative analysis of the GenEval benchmark, accounting for varying thresholds τ and $\alpha = 4$. Latency measurements were conducted with a batch size of 1 on a single GPU. The evaluation of Infinity-2B was performed using rewritten prompts in accordance with the methodology outlined in the official repository.

Model	τ	GenEval↑							Latency (s)↓
		Two Obj.	Position	Color Attri.	Counting	Colors	Sin Obj.	Overall	
Infinity-2B	-	0.8586	0.4175	0.5525	0.6844	0.8431	1.0000	0.7260	2.78
+ SparseVAR	0.4	0.8485	0.4250	0.5625	0.7000	0.8457	1.0000	0.7303	2.64
	0.5	0.8359	0.4250	0.5600	0.6781	0.8351	1.0000	0.7224	1.87
	0.6	0.8409	0.4125	0.5475	0.6812	0.8404	1.0000	0.7204	1.47
	0.7	0.8460	0.4225	0.5475	0.6719	0.8378	1.0000	0.7209	1.36
HART-0.7B	-	0.6919	0.1625	0.2825	0.3688	0.8617	0.9938	0.5602	1.32
+ SparseVAR	0.4	0.7071	0.1450	0.2650	0.3938	0.8777	0.9906	0.5632	1.25
	0.5	0.7045	0.1600	0.2575	0.3969	0.8644	0.9906	0.5623	1.18
	0.6	0.7071	0.1600	0.2825	0.3562	0.8670	0.9906	0.5606	0.99
	0.7	0.6035	0.1200	0.2125	0.3344	0.8351	0.9656	0.5119	0.81

decreasing by only 0.0051 in GenEval and 0.0033 in DPG-Bench. Similarly, when $\tau = 0.6$, SparseVAR achieves a 25% reduction in the inference latency of HART, while exhibiting only a marginal decrease in GenEval(0.0030) and in DPG-Bench(0.0023). These results indicate that SparseVAR effectively preserves the image generation quality of high-frequency regions during high-resolution stages.

Table 2. Quantitative evaluation on DPG-Bench. Latency measurements are conducted on a single GPU using a batch size of 1.

Model	τ	DPG-Bench↑			Latency (s)↓
		Global.	Relation	Overall	
Infinity-2B	-	0.8419	0.9283	0.8289	2.55
+ SparseVAR	0.4	0.8541	0.9246	0.8282	2.34
	0.5	0.8480	0.9242	0.8254	1.69
	0.6	0.8632	0.9237	0.8260	1.35
	0.7	0.8511	0.9270	0.8256	1.20
HART-0.7B	-	0.8710	0.9295	0.8099	1.31
+ SparseVAR	0.4	0.8571	0.9233	0.8092	1.24
	0.5	0.8602	0.9233	0.8082	1.19
	0.6	0.8602	0.9246	0.8069	1.00
	0.7	0.8663	0.9254	0.8072	0.83

Human preference evaluation. To further investigate the impact of SparseVAR's acceleration on human evaluation preferences, we conducted experiments on two datasets focused on human preference assessment. As shown in Table 3, on the HPSv2.1 dataset, when $\tau = 0.7$, SparseVAR reduces the average inference latency of Infinity by 49.43% while the overall score decreases by only 0.47. On the ImageReward dataset, SparseVAR reduces the inference latency by 49.62% with a score reduction of only 0.0266. These results indicate that SparseVAR effectively preserves the quality of high-frequency regions, and the generated images remain aligned with human preferences.

Influence of different α. To investigate the effectiveness of anchor tokens, we conducted comparative experiments on HART and Infinity by varying the grid size α. As shown in Table 4, retaining anchor tokens improves the image generation quality of both HART and Infinity while introducing minimal additional inference overhead. Notably, HART is more significantly affected due to its residual diffusion mechanism, which takes the final stage's output as input. Without logits assigned to the early exiting low-frequency tokens via anchor tokens, residual diffusion lacks these low-frequency inputs, leading to incomplete detail refinement in these regions. Consequently, the strategy of retaining anchor tokens effectively preserves the optimization quality of this diffusion process. When α is small, the image generation quality improves further, but the number of tokens required for inference increases significantly, resulting in slower inference. Considering both quality and efficiency, we set $\alpha = 4$.

Influence of different P. To investigate the impact of early-exited stages on image generation quality, we conducted experiments on GenEval to compare the effects of different values of P on generation quality and average inference latency. As shown in Table 5, when P is set to a small value, meaning the model exits at a very early stage, the image generation quality significantly deteriorates while inference latency is not greatly reduced. This is because next-scale prediction progressively increases the resolution, making later stages much more computationally intensive than earlier ones. Furthermore, earlier stages primarily generate low-frequency content, making early exits more sensitive to the final image quality. Therefore, considering both quality and

Table 3. Quantitative evaluation on Human Preference Metrics. This table provides a detailed quantitative analysis on two human preference benchmarks, considering varying thresholds τ and a fixed local window size of 4. Latency measurements were performed with a batch size of 1 on a single GPU to ensure consistency and accuracy.

Model	τ	ImageReward		HPSv2.1					
		Score↑	Latency(s)↓	Anime	Concept-Art	Paintings	Photo	Overall↑	Latency(s)↓
Infinity-2B	-	0.9212	2.64	31.63	30.26	30.28	29.27	30.36	2.61
+ SparseVAR	0.4	0.9147	2.37	31.58	30.13	30.16	29.22	30.27	2.35
	0.5	0.8969	1.77	31.40	29.95	29.96	29.05	30.09	1.79
	0.6	0.8943	1.42	31.29	29.82	29.77	28.94	29.95	1.40
	0.7	0.8946	1.33	31.21	29.75	29.71	28.88	29.89	1.32
HART-0.7B	-	0.8656	1.32	31.22	29.61	29.10	28.21	29.53	1.30
+ SparseVAR	0.4	0.8818	1.25	31.19	29.58	29.08	28.19	29.51	1.26
	0.5	0.8818	1.20	31.06	29.47	28.96	28.09	29.40	1.19
	0.6	0.8121	1.01	30.25	28.68	28.13	27.51	28.64	1.02
	0.7	0.4333	0.82	27.18	25.60	25.13	24.93	25.71	0.81

Table 4. Evaluation of different local window sizes α on GenEval. τ is set as 0.6. Inference latency is measured with batch size of 1 and in seconds. - means that we do not keep anchor tokens.

α	Infinity		HART	
	Score	Latency(s)	Score	Latency(s)
2	0.7235	1.76	0.5615	1.11
3	0.7210	1.54	0.5578	1.06
4	0.7204	1.47	0.5606	0.99
5	0.7200	1.46	0.5560	0.97
-	0.7190	1.38	0.5502	0.93

Table 5. Impact of different P on image generation quality and inference latency on GenEval. We report the generation quality score (↑) and inference latency in seconds (↓) for each P.

P	Infinity		HART	
	Score	Latency(s)	Score	Latency(s)
6	0.6805	1.29	0.5529	0.98
8	0.7085	1.35	0.5577	0.98
9	0.7126	1.39	0.5565	0.99
10	0.7204	1.47	0.5602	0.99
11	0.7261	1.76	0.5625	1.03
12	0.7274	2.21	0.5607	1.05

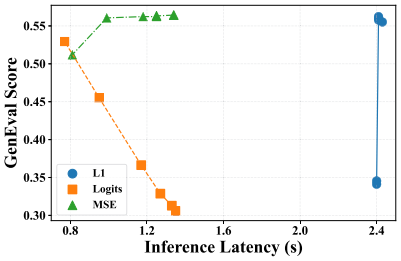

Figure 6. Comparison of different metrics for distinguishing high- and low-frequency regions.

efficiency, we set $P = 10$.

Comparisons with different early exiting metrics. To compare the effectiveness of different metrics for distinguishing high- and low-frequency regions for early exiting, we evaluated three approaches: using the MSE changes in specific block, the logits similarity generated at each stage, and the ℓ_1 differences between images generated by the cumulative residuals decoded at each stage and the previous stage. We have detailed the calculation specifics of the three metrics in the appendix. As shown in Figure 6, using MSE as the metric for early exiting provides the most lightweight and accurate measurement of low-frequency regions, enabling precise early exits. Although using the ℓ_1 difference of generated images can also effectively distinguish high- and low-frequency regions, it requires an additional decoder operation at each stage, resulting in extra computational overhead.

Impact of block selection on MSE-based frequency estimation. To investigate the impact of using MSE changes from different blocks for high- and low-frequency selection on image generation quality, we conducted comparative experiments across various blocks on Infinity with $\tau = 0.6$. As shown in Figure 8, while some blocks exhibit clear distinctions in MSE variations between high and low frequencies, others do not. Among all blocks, the 16^{th} block achieved the best results. Therefore, we utilize the MSE changes from the 16^{th} block for high- and low-frequency estimation. Additional visualizations of MSE changes across different blocks are provided in the appendix.

Qualitative visualizations. To provide a more intuitive demonstration of SparseVAR's ability to accurately reduce computations in low-frequency regions, we visualize the token distribution for early exiting across different stages with $\tau = 0.6$ and $\alpha = 4$. As shown in Figure 7, the visualizations illustrate the sparsity pattern of SparseVAR's early exit process in 11^{th} to 13^{th} stage. As clearly illustrated in the figure, SparseVAR effectively excludes the majority of low-frequency regions, retaining high-frequency regions for inference. By the final stage, only a small number of tokens remain, yet the majority of the image generation quality is preserved. Compared to the baseline, SparseVAR achieves a significant acceleration in inference speed with minimal degradation in image generation quality.

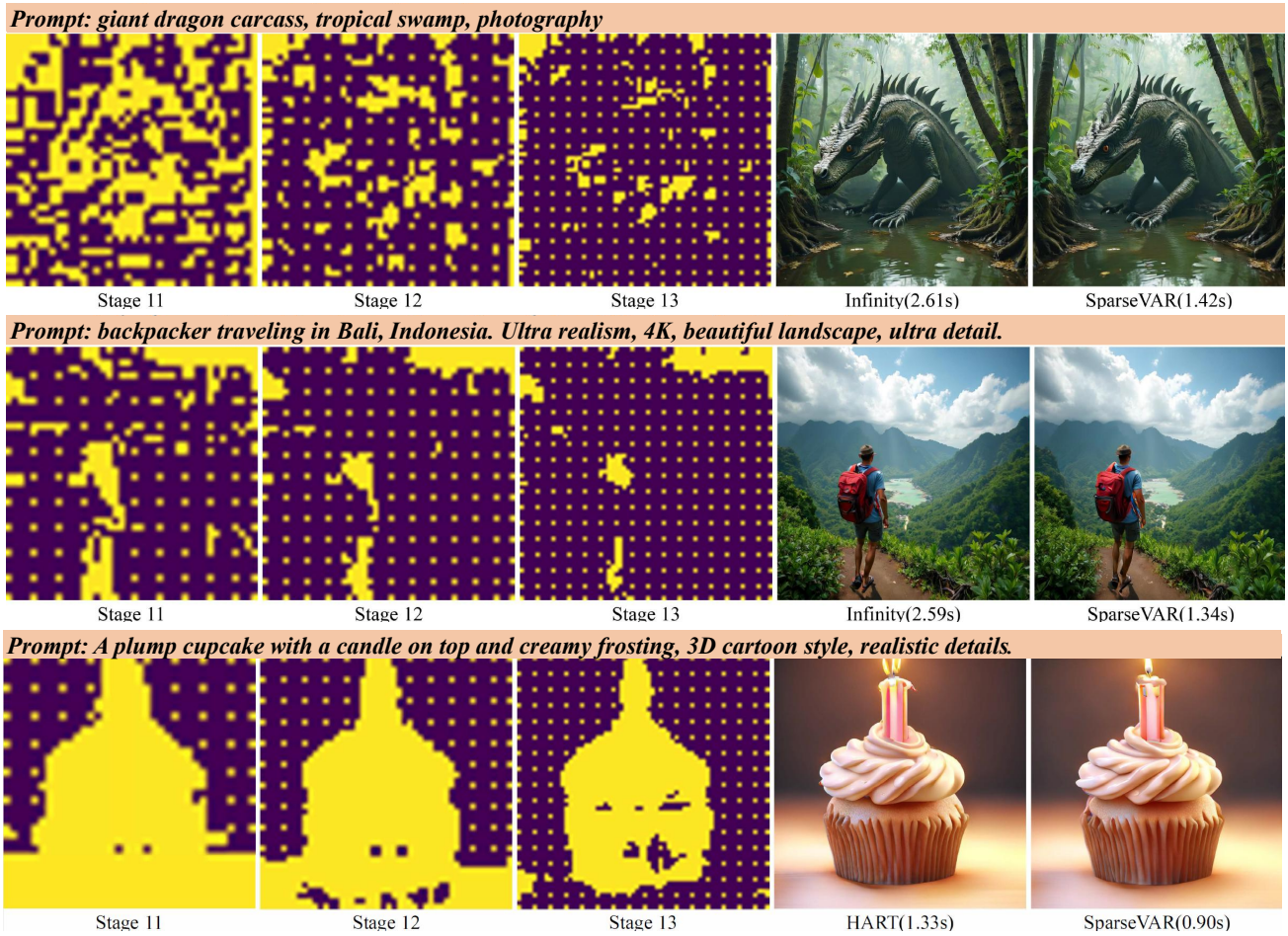

Figure 7. Qualitative visualizations of SparseVAR. The yellow and purple colors represent the tokens identified as retained and early-exited at each stage, respectively.

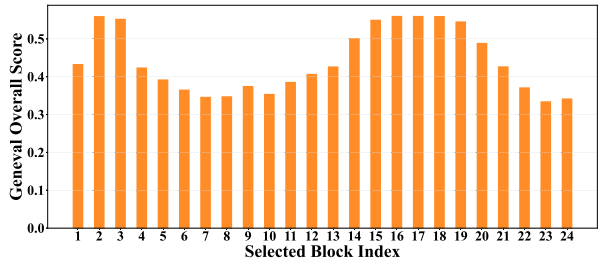

Figure 8. Impact of block selection on MSE-based frequency estimation. Image generation quality is evaluated using the HART-0.7B across different blocks.

6. Conclusion

This paper explores the redundancy of token computations in high-resolution stages of next-scale prediction models and proposes SparseVAR, a novel method for effectively accelerating image generation. The approach offers a simple yet effective strategy that dynamically identifies and excludes low-frequency tokens, requiring no additional training. By leveraging the strong local dependencies between neighboring tokens, SparseVAR significantly reduces computational overhead while preserving image quality. Overall, SparseVAR enhances the efficiency of next-scale prediction models with minimal loss in performance, providing a practical solution for high-resolution image synthesis.

Limitations and future work. Our research primarily focuses on next-scale prediction models, leaving the broader applicability of early exclusion of low-frequency tokens in other autoregressive generation models unexplored. Extending this concept to a wider range of autoregressive frameworks remains an important direction for future work. Additionally, the current approach employs a uniform sampling strategy for anchor selection. However, fixed window frequencies and anchor positions may not be optimal for all images, as the characteristics of low-frequency regions vary significantly across different inputs. Developing a dynamic method to adaptively adjust the sampling frequency and anchor placement based on image content could further enhance the generation quality of low-frequency regions.

7. Acknowledgement

This work was partially supported by the Joint Funds of the National Natural Science Foundation of China (Grant No.U24A20327).

References

[1] Arif, K. H. I., Yoon, J., Nikolopoulos, D. S., Vandierendonck, H., John, D., and Ji, B. Hired: Attention-guided token dropping for efficient inference of high-resolution vision-language models in resource-constrained environments. *arXiv preprint arXiv:2408.10945*, 2024. 2, 3

[2] Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y., et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023. 3

[3] Bolya, D., Fu, C., Dai, X., Zhang, P., Feichtenhofer, C., and Hoffman, J. Token merging: Your vit but faster. In *ICLR*. OpenReview.net, 2023. 2, 3, 11

[4] Chen, J., Yu, J., Ge, C., Yao, L., Xie, E., Wang, Z., Kwok, J. T., Luo, P., Lu, H., and Li, Z. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *ICLR*, 2024. 3

[5] Chen, L., Zhao, H., Liu, T., Bai, S., Lin, J., Zhou, C., and Chang, B. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *ECCV*, pp. 19–35. Springer, 2025. 2, 3

[6] Chen, X., Zhang, Y., Wang, Y., Shu, H., Xu, C., and Xu, C. Optical flow distillation: Towards efficient and stable video style transfer. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pp. 614–630. Springer, 2020. 1

[7] Ding, M., Yang, Z., Hong, W., Zheng, W., Zhou, C., Yin, D., Lin, J., Zou, X., Shao, Z., Yang, H., et al. Cogview: Mastering text-to-image generation via transformers. *Advances in neural information processing systems*, 34:19822–19835, 2021. 1

[8] Esser, P., Rombach, R., and Ommer, B. Taming transformers for high-resolution image synthesis. In *CVPR*, pp. 12873–12883, 2021. 3

[9] Ghosh, D., Hajishirzi, H., and Schmidt, L. Geneval: An object-focused framework for evaluating text-to-image alignment. *NeurIPS*, 36, 2024. 5

[10] Han, J., Liu, J., Jiang, Y., Yan, B., Zhang, Y., Yuan, Z., Peng, B., and Liu, X. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. *arXiv preprint arXiv:2412.04431*, 2024. 1, 3, 5

[11] He, Y., Chen, F., He, Y., He, S., Zhou, H., Zhang, K., and Zhuang, B. Zipar: Accelerating autoregressive image generation through spatial locality. *arXiv preprint arXiv:2412.04062*, 2024. 2

[12] He, Y., Chen, F., Liu, J., Shao, W., Zhou, H., Zhang, K., and Zhuang, B. Zipvl: Efficient large vision-language models with dynamic token sparsification and kv cache compression. *arXiv preprint arXiv:2410.08584*, 2024. 2, 3, 11

[13] Hu, X., Wang, R., Fang, Y., Fu, B., Cheng, P., and Yu, G. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024. 5

[14] Lee, D., Kim, C., Kim, S., Cho, M., and Han, W.-S. Autoregressive image generation using residual quantization. In *CVPR*, pp. 11523–11532, 2022. 1

[15] Lee, D., Kim, C., Kim, S., Cho, M., and Han, W.-S. Autoregressive image generation using residual quantization. In *CVPR*, pp. 11523–11532, 2022. 3

[16] Lee, S.-H., Wang, J., Zhang, Z., Fan, D., and Li, X. Video token merging for long-form video understanding. *arXiv preprint arXiv:2410.23782*, 2024. 2, 3

[17] Li, B., Qi, X., Lukasiewicz, T., and Torr, P. Controllable text-to-image generation. *NeurIPS*, 32, 2019. 1

[18] Li, X., Qiu, K., Chen, H., Kuen, J., Gu, J., Raj, B., and Lin, Z. Imagefolder: Autoregressive image generation with folded tokens. *arXiv preprint arXiv:2410.01756*, 2024. 1

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

[20] Mentzer, F., Minnen, D., Agustsson, E., and Tschannen, M. Finite scalar quantization: Vq-vae made simple. In *ICLR*. 1

[21] Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. SDXL: improving latent diffusion models for high-resolution image synthesis. In *ICLR*, 2024. 3

[22] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation. In *ICML*, pp. 8821–8831. Pmlr, 2021. 1

[23] Razavi, A., Van den Oord, A., and Vinyals, O. Generating diverse high-fidelity images with vq-vae-2. *NeurIPS*, 32, 2019. 3

[24] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. Generative adversarial text to image synthesis. In *ICML*, pp. 1060–1069. PMLR, 2016. 1

[25] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *CVPR*, pp. 10684–10695, 2022. 1

[26] Shang, Y., Cai, M., Xu, B., Lee, Y. J., and Yan, Y. Llava-prumerge: Adaptive token reduction for efficient large

multimodal models. *arXiv preprint arXiv:2403.15388*, 2024. 2, 3

[27] Sun, P., Jiang, Y., Chen, S., Zhang, S., Peng, B., Luo, P., and Yuan, Z. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024. 1

[28] Tang, H., Wu, Y., Yang, S., Xie, E., Chen, J., Chen, J., Zhang, Z., Cai, H., Lu, Y., and Han, S. Hart: Efficient visual generation with hybrid autoregressive transformer. *arXiv preprint arXiv:2410.10812*, 2024. 1, 2, 3, 5

[29] Tian, K., Jiang, Y., Yuan, Z., Peng, B., and Wang, L. Visual autoregressive modeling: Scalable image generation via next-scale prediction. 2024. 1, 2, 3, 11

[30] Van Den Oord, A., Vinyals, O., et al. Neural discrete representation learning. *NeurIPS*, 30, 2017. 3

[31] Wang, C., Wang, C., Xu, C., and Tao, D. Tag disentangled generative adversarial networks for object image re-rendering. In *International joint conference on artificial intelligence (IJCAI)*, 2017. 1

[32] Wang, X., Zhang, X., Luo, Z., Sun, Q., Cui, Y., Wang, J., Zhang, F., Wang, Y., Li, Z., Yu, Q., et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024. 1

[33] Wu, X., Hao, Y., Sun, K., Chen, Y., Zhu, F., Zhao, R., and Li, H. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023. 5

[34] Xu, J., Liu, X., Wu, Y., Tong, Y., Li, Q., Ding, M., Tang, J., and Dong, Y. Imagereward: Learning and evaluating human preferences for text-to-image generation. *NeurIPS*, 36, 2024. 5

[35] Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., and He, X. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*, pp. 1316–1324, 2018. 1

[36] Yu, J., Li, X., Koh, J. Y., Zhang, H., Pang, R., Qin, J., Ku, A., Xu, Y., Baldridge, J., and Wu, Y. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021. 3

[37] Yu, L., Lezama, J., Gundavarapu, N. B., Versari, L., Sohn, K., Minnen, D., Cheng, Y., Gupta, A., Gu, X., Hauptmann, A. G., et al. Language model beats diffusion-tokenizer is key to visual generation. In *ICLR*. 1

[38] Yu, L., Lezama, J., Gundavarapu, N. B., Versari, L., Sohn, K., Minnen, D., Cheng, Y., Gupta, A., Gu, X., Hauptmann, A. G., Gong, B., Yang, M., Essa, I., Ross, D. A., and Jiang, L. Language model beats diffusion - tokenizer is key to visual generation. In *ICLR*, 2024. 3