Depth Adaptive Efficient Visual Autoregressive Modeling

Chunliang Li[1*] Tianze Cao[1*] Sanyuan Zhao[1,2†]
[1]Beijing Institute of Technology [2]Beijing Institute of Technology, Zhuhai
jbjic@icloud.com {caotianze, zhaosanyuan}@bit.edu.cn

Abstract

Visual Autoregressive (VAR) modeling inefficiently applies a fixed computational depth to each position when generating high-resolution images. While existing methods accelerate inference by pruning tokens using frequency maps, their binary hard-pruning approach is fundamentally limited and fails to improve quality even with better frequency estimation. Observing that VAR models possess significant depth redundancy, we propose a paradigm shift from pruning entire tokens to adaptively allocating per-token computational depth. To this end, we introduce DepthVAR, a training-free framework that dynamically allocates computation. It integrates an adaptive depth scheduler, which assigns computational depth via a cyclic rotated schedule for balanced, non-static refinement, with a dynamic inference process that translates these depths into layer-major masks, selectively applies transformer blocks, and blends the resulting codes to ensure each token's influence is proportional to its processing depth. Extensive experiments show that DepthVAR achieves 2.3×-3.1× acceleration with minimal quality loss, offering a competitive compute-performance trade-off compared to existing hard-pruning approaches. Code is available at https://github.com/STOVAGtz/DepthVAR.

1. Introduction

Recent advances in Autoregressive (AR) models use 'next-token' prediction to enable the step-by-step synthesis of complex visual structures, offering a unified probabilistic framework for text-to-image generation. However, the sequential nature of AR modeling leads to long token chains, which increases computational cost and memory demands, especially for high-resolution images. Visual Autoregressive (VAR) modeling [25, 54, 58] mitigates this issue by shifting to 'next-scale' prediction and generating images hierarchically across scales. This substantially reduces sequence length and prediction latency, yet tokens grow quadratically with each scale, creating significant computational overhead from the inefficient uniform processing of tokens that represent regions requiring less computation.

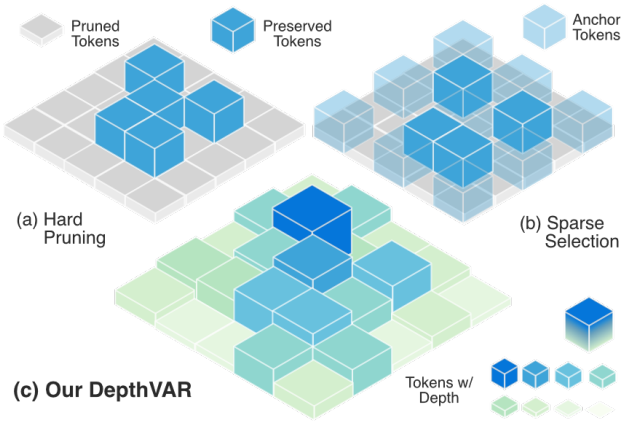
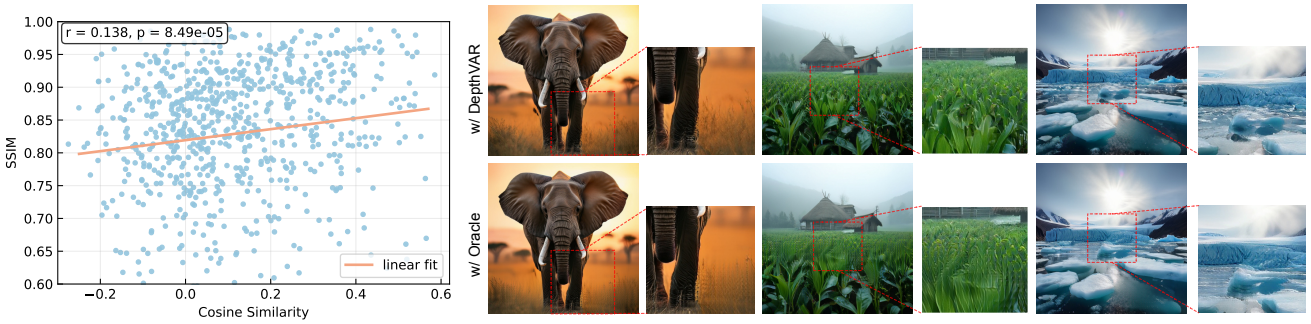


Figure 1. **Comparison of VAR acceleration paradigms.** (a) Hard token pruning (e.g. [24]) discards tokens. (b) Sparse token selection (e.g. [5]) retains anchor tokens to preserve background structure. (c) We adaptively vary the layers processed per token.

In pursuit of non-uniform processing for efficient VAR inference, prior works [5, 24] prune tokens at larger scales, assuming high-frequency tokens are more critical for later-stage refinement. These methods estimate high-frequency distributions from intermediate outputs to identify and discard less important tokens. However, such hand-crafted frequency estimations are often inaccurate, degrading quality in pruned regions, which is sometimes mitigated with sparse background grids [5]. Specifically, as shown in Fig. 2, we find that more accurate frequency estimation does not guarantee improved generation quality, suggesting a fundamental limitation in the hard-pruning paradigm.

These frequency-based methods [5, 24] dichotomize tokens into a 'keep' or 'prune' status. This motivates our investigation into a more continuous form of computational scaling, where we find that VAR models exhibit exploitable token-wise and layer-wise depth redundancy, diverging from previous findings [38], which suggests that reducing per-token depth rather than hard-pruning tokens can potentially save computation while better preserving image qual-

*Equal contribution
†Corresponding author



(a) Frequency approximation vs. image quality.

(b) Qualitative results of hard-pruning with an oracle frequency mask.

Figure 2. **Limitations of frequency-based pruning.** (a) The accuracy of frequency map approximation, a common heuristic for token pruning, correlates poorly with final image quality. (b) Employing a perfect oracle frequency mask for hard-pruning still results in significant quality degradation, which points to an inherent flaw in the strategy.

ity. While depth redundancy [11, 14, 16] motivates modern early exiting [16, 47] and dynamic depth [11, 26, 43] methods, differences in the saturation behavior make direct transfer to VAR models non-trivial. Accordingly, we hypothesize that computational resources should be allocated in a more continuous manner across tokens, allowing for a non-binary approach to non-uniform processing at larger scales.

Based on these observations, we propose DepthVAR, a training-free inference acceleration framework that precisely controls per-token inference depth. As illustrated in Fig. 1, DepthVAR extends beyond prior hard token pruning and sparse selection paradigms by introducing a continuous depth allocation scheme. To exploit depth redundancy, our dynamic inference framework converts token depth scores into a layer-major mask via bit-reversal to ensure unbiased layer utilization, selectively applies transformer blocks while reusing cached layer behaviors to maintain continuity, and blends the resulting codes to ensure each position's influence is proportional to its processing depth. These per-position depths are determined by an Adaptive Depth Score Scheduler, which applies a cyclic rotated schedule to prior decision rank maps. This process generates non-static depth scores to ensure balanced, continuous refinement across the image. Integrating these components, DepthVAR adaptively achieves efficient and fine-grained dynamic inference. We demonstrate that our DepthVAR can accelerate VAR inference with a better performance trade-off than previous hard prune approaches and achieves 2.3×-3.1× acceleration with minimal quality loss.

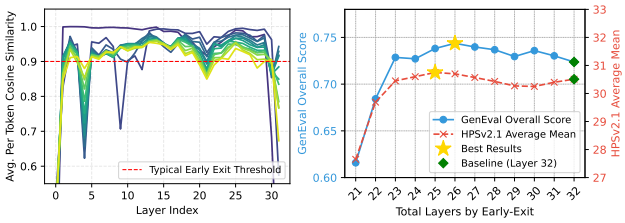
In a nutshell, our contributions are as follows,

- We reveal the limitations of frequency-based hard-pruning and identify exploitable depth redundancy in VAR models as a more effective path toward acceleration.
- We propose DepthVAR, a training-free framework that enables continuous depth allocation by integrating a cyclic adaptive depth scheduler with a dynamic inference mechanism for selective computation and code blending.
- Experiments on multiple benchmarks demonstrate that DepthVAR achieves 2.3×-3.1× acceleration and marginally superior quality compared with prior hard-pruning acceleration methods.

2. Related Work

Dynamic Depth in Transformers. A key strategy for accelerating Transformer inference [7, 56, 59] is to dynamically activate input-specific sub-networks, rather than static methods like quantization [1, 39, 51, 53] or distillation [23, 28, 45], including Mixture-of-Experts (MoE) [18, 34, 50] and adaptive depth methods [10, 13, 16, 47] that vary layer usage per token. As a form of adaptive depth, early exiting was first introduced for DNNs and CNNs [2, 30, 57], and was later applied to encoder-only Transformers like BERT [12, 29, 48, 62, 64, 65], where most methods rely on model confidence scores or small exit classifiers [46, 48, 62], and subsequently extended to decoder-based LMs [11, 14, 43, 47, 49]. Specifically, Universal Transformer [10] introduced position-wise stopping with ACT [21], while Depth-Adaptive Transformer [13] trained auxiliary exits. Furthermore, LayerDrop [16] demonstrated that dropping layers during training enables inference over sub-networks, and CALM [47] aligns token-level exit decisions with sequence quality targets. More flexible approaches learn to allocate depth per token [11, 17, 26, 43], e.g., Mixture-of-Depths(MoD) [43] trained block routers to bypass certain blocks, later extended by Router-Tuning [26] to a retro-fit framework. Methods like SkipDecode [11] and Depth Decay Decoding [17] determine depths via decay rules, offering more flexibility over early-exiting. The emergence of similar strategies in vision models [19, 38, 55] further underscores their potential for accelerating VARs.

Efficient Visual Autoregressive Modeling. The scale-by-scale generation paradigm of Visual Autoregressive Modeling (VAR) [4, 25, 54, 58] prevents the direct applica-



(a) Token-wise Layer Similarity (b) Early-exiting Evaluations

Figure 3. **Evidence of depth redundancy in pretrained VAR models.** (a) Token-wise representation similarity between consecutive layers shows saturation at different depths (darker colors for smaller scales). (b) Generation quality peaks before the final layer with early exiting, confirming full depth is not always optimal.

tion of parallel decoding strategies from sequential generation, such as Speculative Decoding [35] and Block-wise Parallel Decoding [52]. Theoretically, VAR improves time complexity from $O(n^6)$ to $O(n^4)$ [58], with a potential near-quadratic limit of $O(n^{2+o(1)})$ [31], and LiteVAR [61] achieves near $O(n^2)$ in practice. Some approaches reduce memory [37] or employ linear complexity mechanisms like Mamba [22] to decouple inter-scale computations [44]. Inspired by Speculative Decoding [35], CoDe [6] uses a large and a small model to reduce redundant computations on larger scales. FastVAR [24] and SparseVAR [5] leverage the frequency characteristics of scale prediction to reduce computation by lowering the coefficient α in the time complexity $O(\alpha \times n^4)$, and SkipVAR [36] further explores this by exploiting high-frequency differences across different samples. Architectural improvements to VAR, such as HMAR [33] and HART [54], require extra training or structural changes. To the best of our knowledge, the closely related FreqExit [38] is a training-time early-exiting method. In contrast, our approach is training-free and dynamically adjusts the network depth token-wise during inference.

3. Methodology

3.1. Empirical Observations

Limitations in Frequency Approximation. Previous works [5, 24] accelerate inference by pruning low-frequency tokens identified via approximated frequency maps, based on the observation that different frequency components converge at different rates during generation. Contrary to intuition, our empirical analysis shows that more accurate frequency map approximations do not guarantee better image quality. On hard-pruning [5], we evaluated 800 samples by comparing its predicted frequency mask with a Sobel-filtered ground-truth. As shown in Fig. 2a, the approximation accuracy exhibits only a weak positive correlation with final image quality (SSIM; Pearson's r = 0.138), indicating that refining frequency maps does not reliably improve results. To further investigate, we conducted an oracle experiment using a perfect frequency mask derived directly from the ground-truth image. As shown in Fig. 2b, even this ideal hard-pruning degrades quality, revealing a deeper issue with the assumption that low-frequency regions can be safely omitted. These observations suggest that binary hard-pruning is inherently limited, motivating a shift towards more granular computational allocation strategies.

Depth Redundancy in Pretrained VAR. To improve generalization and prevent overfitting, VAR models [25, 54, 58] often employ LayerDrop [16] during training. This regularization strategy introduces depth redundancy [14, 16] into the pretrained model, which can be leveraged for faster inference. To confirm this redundancy, we follow [25] and measure performance when forcing all tokens to exit at earlier layers. As shown in Fig. 3b, generation quality on two benchmarks peaks before the final layer rather than increasing monotonically, indicating that the model is overparameterized in depth and can be accelerated by trimming layers without harming quality. We further examine token-wise similarity across consecutive layers (Fig. 3a) and observe that token representations saturate at some layers, revealing token-specific depth redundancy. This contrasts with prior findings [38] based on cosine-similarity saturation used in classical early-exit methods [47], suggesting that VAR models exhibit a distinct form of redundancy that may require tailored exploitation strategies. These observations imply that simpler tokens may not benefit from full-depth processing, motivating our approach to adaptively reduce layer depth to better harness this property.

3.2. Preliminaries

At a high level, the visual autoregressive modeling (VAR) [25, 54, 58] predicts K multi-scale residual maps $(r_0, r_1, \ldots, r_{K-1})$ and accumulates the upscaled residual maps to gain a feature map f_i at each scale $i \in \{0, \ldots, K-1\}$. With encoded start token r_0 from text prompts [25, 40, 54], given a VAR model with L stacked transformer layers [59], the parallel prediction of $h_i \times w_i$ tokens in $r_i \in [V]^{h_i \times w_i}$ starts from scale-conditioned downsampled feature embeddings:

$$r_i = \text{Layer}_{0\ldots L}(\text{embed}(\textit{down}(f_{i-1}))) \tag{1}$$

where *down* is bilinear downsampling [25] and $\text{Layer}_{0\ldots L}$ is the cumulative form of using $r_i^\ell = \text{Layer}_\ell(r_i^{\ell-1})$, where Layer_ℓ is the ℓ-th block. Upsampling retrieved features z_i from the codebook at the predicted logits $p_i = \text{head}(r_i)$ gives the intermediate feature maps:

$$f_i = f_{i-1} + \textit{up}(z_i, h_{K-1}, w_{K-1}) \tag{2}$$

where $z_i = \text{lookup}(p_i)$. In the last prediction step, f_{K-1} is used for generating the prediction image. This standard prediction process utilizes all model layers and naturally allocates equal computation for each position on the image.

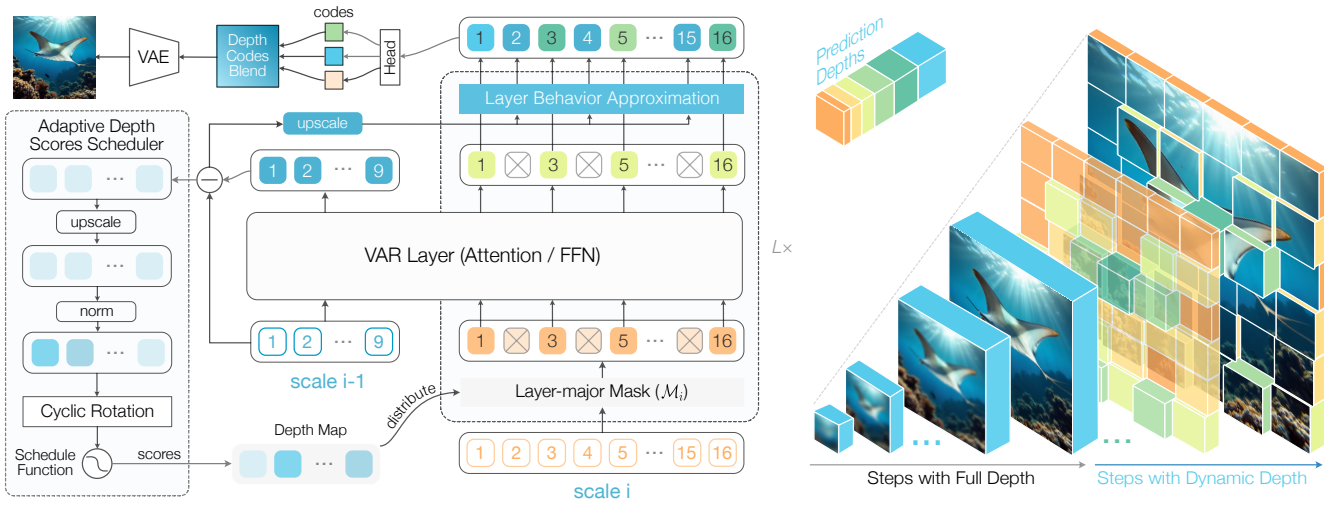


Figure 4. **Overview of dynamic depth inference in DepthVAR. Left:** At each scale i, we first use Adaptive Depth Score Scheduler to generate adaptive depth scores $\mathcal{S}_i$ using layer-wise changes from the previous scale, which are converted to a layer-major mask $\mathcal{M}_i$ via bit-reversal. The dynamic depth inference performs masked prediction using $\mathcal{M}_i$, reinstating cached layer behaviors from the last scale, and blends the resulting codes based on $\mathcal{S}_i$ to produce each scale's final output. **Right:** Early steps follow standard inference, while dynamic depth is applied in later stages.

3.3. DepthVAR: Predicting with Dynamic Depth

As shown in Fig. 4, our key intuition is to adaptively reduce computational depth by assigning each token map position a specific score, which are translated into layer-major masks to selectively activate transformer blocks, while masked positions are restored and resulting codes are blended to ensure each token's influence is proportional to its processing depth. Specifically at scale i, given precomputed depth scores $\mathcal{S}_i \in [0,1]^{h_i \times w_i}$ (Sec. 3.4), we first obtain the integer depth map $\mathcal{D}_i = \lfloor \mathcal{S}_i \cdot L \rfloor$ to guide the mapping process.

Depth Map to Layer-major Mask by Bit-reversal. We then permute position-wise depths $\mathcal{D}_i \in \mathbb{N}^{h_i \times w_i}$ to layer-major masks $\mathcal{M}_i \in \{0,1\}^{L \times h_i \times w_i}$ for easier computation, by distributing per-token depths uniformly across layers for each position (m,n) to achieve unbiased layer utilization. This prevents layers of shallower tokens from being disproportionately pruned. Let $k = \lceil \log_2 L \rceil$, and we define the bit-reversal permutation $\pi_L : \{0,\dots,L-1\} \to \{0,\dots,L-1\}$ by:

$$\pi_L(x) \;=\; \mathtt{rev}_k(x) \tag{3}$$

where $x = \sum_{j=0}^{k-1} b_j 2^j$, $\mathtt{rev}_k(x) = \sum_{j=0}^{k-1} b_j 2^{k-1-j}$. Bit-reversal is also the index ordering used in radix-2 Cooley–Tukey FFTs [8, 15]. For each token index $(m,n) \in [0,h_i) \times [0,w_i) \cap \mathbb{N}^2$, we choose the set of active layers as $\mathcal{L}_i(m,n) \;=\; \{\,\pi_L(x)\,\}_{x=0}^{d_i(m,n)-1}$ and define the layer-major binary mask $\mathcal{M}_i \in \{0,1\}^{L \times h_i \times w_i}$ as:

$$\mathcal{M}_i(\ell,m,n) = \mathbf{1}\{\ell \in \mathcal{L}_i(m,n)\}\,. \tag{4}$$

For example, with $L = 32$ and $d_i(m,n) = 5$, one obtains $\mathcal{L}_i(m,n) = \{0,16,8,24,4\}$.

Masked Layer Behavior Approximation. At each layer ℓ, we reduce computation by processing only the active positions defined by the spatial mask slice $\mathcal{M}_i(\ell)$, and restore the cached proxy from the last scale at masked positions:

$$\begin{aligned} r_i^\ell = &\underbrace{\mathbf{Layer}_\ell(r_i^{\ell-1} \odot \mathcal{M}_i(\ell))}_{\text{sparse prediction}} + \\ &\underbrace{up(r_{i-1}^\ell - r_{i-1}^{\ell-1}, h_i, w_i) \odot (1 - \mathcal{M}_i(\ell))}_{\text{cached proxy restoration}} \end{aligned} \tag{5}$$

where $up(\cdot,\cdot,\cdot)$ is the bilinear upscale. Eq. (5) ensures that subsequent layers receive a spatially complete feature map, allowing the upscaled residuals cached from the previous scale to serve as a consistent and continuous layer behavior proxy for masked regions. We use the original positions (m,n) as embedding positions in each layer block's RoPE2d [25, 27], and restore $(1 - \mathcal{M}_i(0))$ by the similarity criteria following [5] after the last layer $\ell = L-1$, empirically minimizing the impact of masked tokens omitted from the attention context by exploiting inter-scale local stability.

Depth-based Code Blending. Finally, to let the residual added to the intermediate feature map be affected more by deeper tokens and less by shallower ones, we reweight the predicted codes z_i by the depth scores map $\mathcal{S}_i = \big[\big[s_i(m,n)\big]_{m=0}^{h_i-1}\big]_{n=0}^{w_i-1}$ as $z_i = \mathcal{S}_i \cdot \text{lookup}(p_i)$, where $p_i = \text{head}(r_i^L)$. This ensures that the contribution of each token is proportional to its computational investment.

3.4. Adaptive Depth Scores Scheduler

This dynamic depth inference requires depth scores $\mathcal{S}_i \in [0,1]$ estimated for each token position. Directly porting

Table 1. Quantitative evaluation on GenEval. Our DepthVAR achieves a superior trade-off between semantic consistency and inference latency compared to the baseline and other acceleration methods. ‡: requires additional training; *: w/o last two scales; EE: early exit.

Methods	GenEval ↑							Avg Lat.(ms) ↓	Acc. Steps
	Counting	Color Attr.	Two Obj.	Colors	Position	Sin Obj.	Overall		
Infinity [25]	0.6812	0.5375	0.8636	0.8298	0.4300	1.0000	0.7237	2706	/
Infinity-*EE-26* [25]	0.6875	0.5925	0.8561	0.8511	0.4750	1.0000	0.7437	2232	0-12
+ ToMe {0.5, 0.5} [3]	0.6562	0.4050	0.7854	0.7287	0.4100	1.0000	0.6642	1284	11-12
+ SparseVAR-0.7 [5]	0.6812	0.5500	0.8131	0.8378	0.4425	1.0000	0.7208	1281	10-12
+ SkipVAR‡@0.84 [36]	0.7188	0.5375	0.8460	0.8431	0.3975	1.0000	0.7238	1325	10-12
+ FastVAR* [24]	0.7000	0.5525	0.8359	0.8245	0.4300	1.0000	0.7238	1080	9-10
+ DepthVAR*($\mathcal{R}$=7)	0.7188	0.5600	0.8157	0.8245	0.4050	1.0000	0.7207	869	8-10
+ DepthVAR($\mathcal{R}$=7)	0.6906	0.5575	0.8359	0.8271	0.4425	1.0000	0.7256	1168	8-12
+ DepthVAR($\mathcal{R}$=8)	0.6812	0.5725	0.8333	0.8324	0.4375	1.0000	0.7262	1295	9-12
+ DepthVAR($\mathcal{R}$=9)	0.6969	0.5700	0.8333	0.8457	0.4450	1.0000	0.7318	1622	10-12
HART[54]	0.3594	0.2550	0.7197	0.8644	0.1475	0.9875	0.5556	1102	/
HART-*EE-21*[54]	0.3688	0.2550	0.7803	0.8590	0.1575	0.9750	0.5659	987	0-13
+ SparseVAR-0.7[5]	0.3625	0.2200	0.6364	0.8484	0.1275	0.9750	0.5283	636	10-13
+ FastVAR *w/o FlashAttn*[24]	0.3688	0.2175	0.6995	0.8378	0.1200	0.9750	0.5364	1195	12-13
+ DepthVAR($\mathcal{R}$=9)	0.3656	0.2525	0.6869	0.8457	0.1450	0.9781	0.5456	710	10-13
+ DepthVAR($\mathcal{R}$=10)	0.3906	0.2750	0.7197	0.8697	0.1425	0.9875	0.5642	856	11-13

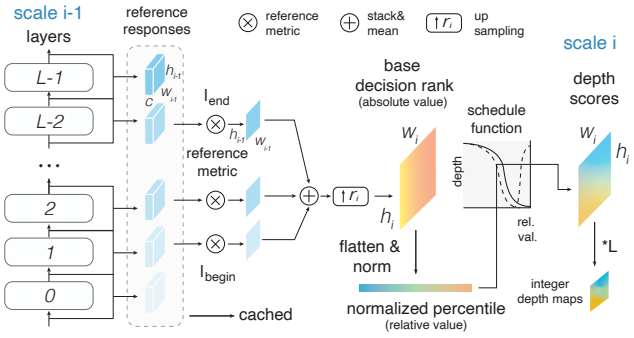


Figure 5. **Adaptive Depth Score Scheduler Pipeline.** Feature changes from scale $i-1$ are aggregated, upsampled, and normalized into percentiles, which are then mapped via a schedule function to continuous depth scores for scale i.

language modeling [47] confidence scores is non-trivial: The visual code space is too large for reliable top-k softmax estimation; regions may visually saturate while hidden-state similarities still vary; and additional classifiers require extra fine-tuning. Prior approaches [5, 24] approximate frequency rank maps either via MSE [5] or mean-subtraction [24]. However, as we showed in Fig. 2, the issue lies more in how these ranks are used rather than the precision of the frequency approximation itself.

Hence, we propose to interpret the previous $(i-1)$-th scale's layer-wise changes as 'past decisions' that guides the current refinement. As illustrated in Fig. 5, this process involves aggregating absolute feature responses into a decision rank map, normalizing these into relative percentiles, and applying a schedule function to map importance to depth. Specifically, given a reference metric $\mathcal{E}$ and a reference range $[\ell_{\text{begin}}, \ell_{\text{end}}]$, we aggregate and upsample reference responses to form a **base decision rank map**:

$$\mathcal{B}_i = up\Big(\sum_{\ell=\ell_{\text{begin}}}^{\ell_{\text{end}}} \mathcal{E}(r_{i-1}^{\ell} - r_{i-1}^{\ell-1}), h_i, w_i\Big) \in \mathbb{R}^{h_i \times w_i}, \quad (6)$$

and compute normalized decision rank percentiles by $\rho_i(m,n) = \frac{1}{h_i w_i} \sum_{(p,q)} \mathbf{1}\{\mathcal{B}_i(p,q) > \mathcal{B}_i(m,n)\} \in [0,1]$.

With a monotonically decreasing **schedule function** $\mathcal{G} : [0,1] \to [0,1]$, we can convert decision rank percentiles to decayed depth scores $s_i(m,n) = \mathcal{G}(\rho_i(m,n))$. To further generalize, we modify $\mathcal{G}$ with a cyclic percentile rotation of magnitude $\eta \in (0,1)$,

$$\mathcal{G}'(\rho) = \begin{cases} \mathcal{G}(\frac{\rho}{\eta}) & 0 \le \rho \le \eta, \\ \mathcal{G}(\frac{1-\rho}{1-\eta}) & \eta < \rho \le 1. \end{cases} \quad (7)$$

which prevents repeatedly updating the same tokens, as previously lowest ranks are rotated away. By applying the rotated mapping on the normalized rank percentiles, we obtain the adaptive **depth scores** as $\mathcal{S}_i = \mathcal{G}'(\rho_i)$.

This percentile-based adaptive scheduling resembles dynamic depth transformers [13, 21, 42], and we extend it across multiscale predictions. By aligning the integral area of $\mathcal{G}'$ on a reference scale $r_{\mathcal{R}}$ with index $\mathcal{R}$, we constrain computation allocated for larger scales and achieve earlier computation reduction than others. In practice, we employ parameter controlled functions $\mathcal{G}_{\text{sigmoid}} = \frac{1}{1+e^{k(x-c)}}$, $\mathcal{G}_{\text{linear-a}} = 1 - c \cdot x$ and $\mathcal{G}_{\text{linear-b}} = c(x-1)$, where k is a user-defined parameter and c is solved from a given integral area. For the reference metric, we use $\mathcal{E}_{\text{MAE}} = |\cdot|, \mathcal{E}_{\text{MSE}} = |\cdot|^2$, and $\mathcal{E}_{\text{sub}} = \cdot - \text{mean}(\cdot)$, we find these metrics behave similarly, while $\mathcal{E}_{\text{MAE}}$ offering the best overall balance.

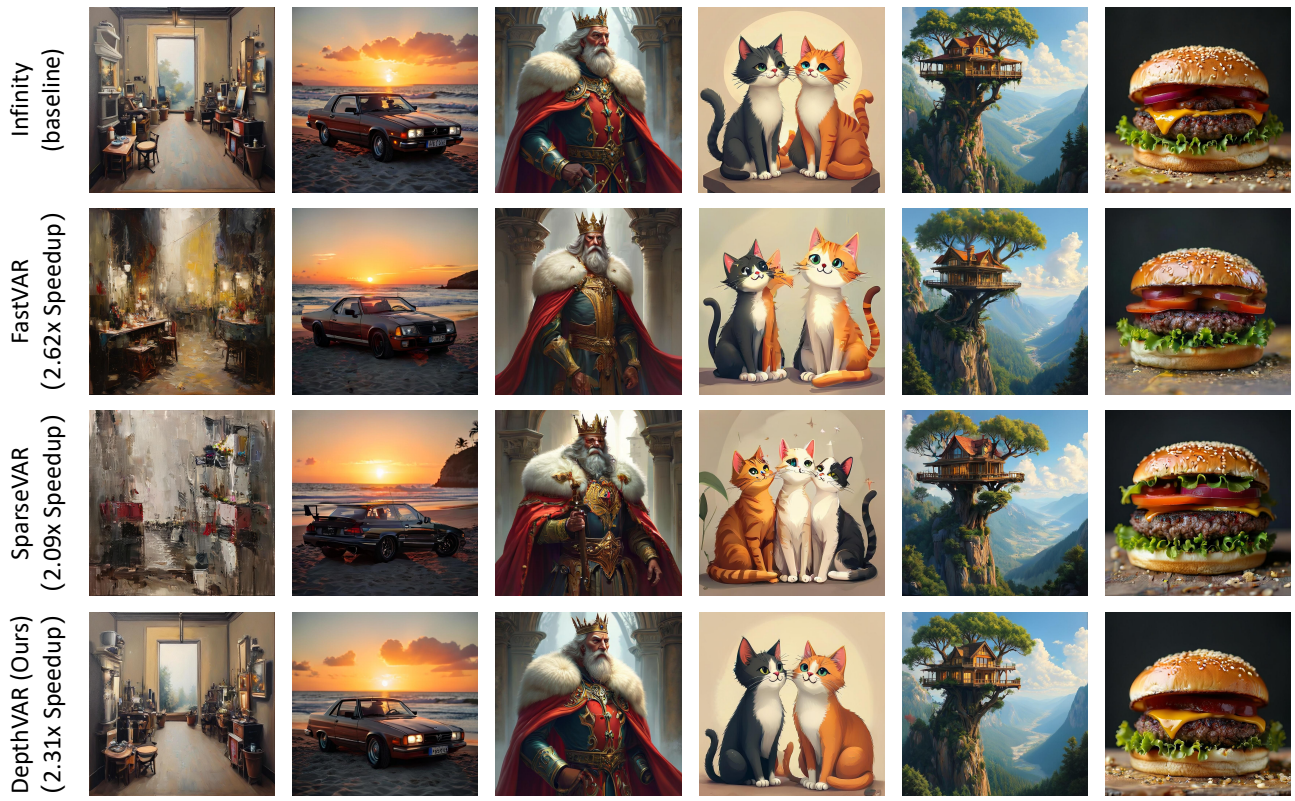


Figure 6. Qualitative visual comparisons between our method, the baseline, and other approaches with relatively fixed inference latency. Our method achieves a 2.31× acceleration while preserving semantic consistency and delivering rich visual details.

4. Experiments

4.1. Experimental Setup

Models and Metrics. We apply DepthVAR on Infinity-2B [25] and HART-0.7B [54], both are capable of generating images up to 1024×1024 high-resolution under text prompts. We evaluate them across multiple popular benchmarks: GenEval [20] for measuring high-level semantic consistency, while HPSv2.1 [60] and ImageReward [63] for human-preference alignment. Unless otherwise specified, we conduct ablation studies on Infinity [25] for consistency and computational efficiency, while HART [54] is used solely for the main results. The performance of each acceleration method is quantified by reporting both its benchmark scores and its wall-clock runtime. To ensure a fair comparison, all methods' hyperparameters are kept as their default settings strictly for evaluation.

Implementation Details. For the reference metric, we use MAE, with the reference range by $\ell_{\text{begin}}, \ell_{\text{end}} = 3, 19$ for Infinity and $8, 17$ for HART. This range is chosen empirically with reference to layers that only attend to the foreground details [5]. We choose the sigmoid as the scheduling function $\mathcal{G}$ by setting $k = 12$, and set $\eta = 0.8$ for cyclic percentile rotation. We apply a similarity threshold of 0.9 at a window of 5 for the $1 - \mathcal{M}_i(0)$ restoration. For reference scale constrained compute, we align $\int_0^\eta \mathcal{G}'(\rho_i)d\rho_i, \forall i > \mathcal{R}$ to $\frac{h_\mathcal{R} \times w_\mathcal{R}}{h_i \times w_i}$, where $\mathcal{R}$ is the reference scale where later scales' computation is limited to, and we choose $i \in \{7, 8, 9\}$ for Infinity and $i \in \{9, 10\}$ for HART. This enables DepthVAR to reduce computation earlier than other methods by up to two scales. Following previous practices [5, 24, 36], FlashAttn [9] is applied to Infinity and not to HART, and the shared VAE latency cost is excluded from speed measurements. All experiments are conducted on a single NVIDIA RTX 3090 GPU with 24GB memory.

4.2. Main Results

Comparison on GenEval. We first evaluate DepthVAR across Infinity [25] and HART [54] on the GenEval benchmark [20], comparing it with hard-pruning, token-merging, and early-exit baselines [3, 5, 24, 25, 54]. As shown in Table 1, DepthVAR consistently achieves superior speed-quality trade-offs: on Infinity, it reaches a 2.3×-3.1× speedup (1168 or 869ms) with negligible quality loss, while on HART ($\mathcal{R}$=10), it improves the overall score by 1.5% with $\sim 1.3\times$ acceleration. Notably, while global early exit strategies [25, 54] can improve scores, they offer limited

Table 2. Quantitative evaluation on Human Preference Metrics, including HPSv2.1 and ImageReward. DepthVAR demonstrates a strong balance between performance and efficiency. ‡: requires additional training; *: w/o last two scales; EE: early exit.

Methods	HPSv2.1						ImageReward	
	Anime	Photo	Painting	Concept-Art	Overall↑	Latency (ms)↓	Score↑	Latency (ms)↓
Infinity [25]	31.68	29.39	30.44	30.36	30.47	2724	0.9515	2716
Infinity-*EE-26* [25]	32.06	29.76	30.48	30.53	30.70	2210	0.8965	2206
+ ToMe {0.5, 0.5}	28.65	26.46	27.12	27.10	27.33	1330	0.7840	1287
+ SparseVAR-0.7 [5]	31.03	28.74	29.57	29.68	29.76	1332	0.8936	1301
+ SkipVAR‡@0.84 [36]	31.60	29.19	30.43	30.27	30.37	1692	0.9376	1744
+ FastVAR* [24]	31.08	28.82	29.97	29.86	29.93	1027	0.9116	1036
+ DepthVAR*($\mathcal{R}$=7)	31.20	28.95	29.94	29.85	29.98	882	0.8996	876
+ DepthVAR($\mathcal{R}$=7)	31.33	29.90	30.03	28.99	30.06	1185	0.9088	1174
+ DepthVAR($\mathcal{R}$=8)	31.42	29.97	30.15	29.10	30.16	1285	0.9171	1303
+ DepthVAR($\mathcal{R}$=9)	31.52	30.12	30.27	29.25	30.29	1625	0.9254	1616
HART [54]	31.30	28.19	29.04	29.56	29.52	1109	0.9013	1103
HART-*EE-21* [54]	31.54	28.25	29.41	29.84	29.76	989	0.9004	985
+ SparseVAR-0.7 [5]	27.69	25.33	25.60	26.12	26.18	669	0.5737	679
+ FastVAR *w/o FlashAttn* [24]	28.66	26.08	26.95	27.31	27.25	1208	0.7448	1209
+ DepthVAR($\mathcal{R}$=9)	28.44	25.72	26.44	26.63	26.81	729	0.6573	727
+ DepthVAR($\mathcal{R}$=10)	29.95	27.04	27.72	28.10	28.20	885	0.7909	880

1.1×-1.2× speedup. Unlike prior methods which suffer semantic instability at higher speedups, DepthVAR leverages exploitable depth redundancy (Fig. 3b) to maintain fidelity under constrained compute, and demonstrates competitive robustness and efficiency as a training-free framework.

Comparison on HPSv2.1 and ImageReward. We evaluate human-preference alignment using HPSv2.1 [60] and ImageReward [63], with results presented in Table 2. DepthVAR shows a strong balance between quality and efficiency; on Infinity [25] it remains highly competitive with the baseline, while on HART [54], it proves more robust without suffering dramatic score collapses. Overall, DepthVAR provides more flexible speed-quality trade-offs than SparseVAR [5] and FastVAR [24], and offers a training-free solution with consistent runtime compared to SkipVAR [36] which requires extra training and has variable latency.

Qualitative Visualizations. As shown in Figure 6, we provide qualitative comparisons between DepthVAR, the baseline, and other acceleration methods at similar latency points. Our approach preserves high image quality and strong semantic consistency with the original generation, delivering rich visual details while achieving a 2.3× speedup. This visual evidence further demonstrates its superiority over existing hard-pruning approaches.

4.3. Ablation Studies

Impact of Depth-based Code Blending. To investigate whether tokens from deeper or shallower layers should contribute differently to image refinement, we conduct ablation experiments on the codes blending strategy across different reference scales. As shown in Tab. 4, with depth-based codes blending disabled—i.e., when the current-scale codes residual prediction is directly added to the feature map—the model performance drops by 0.002-0.015. Rather than contributing equally, this result suggests that deeper tokens should drive the primary updates, while shallower tokens provide fine-grained adjustments.

Table 3. Ablation study of schedule functions on GenEval. Result scores are reported under compute constraints controlled by R.

Methods	Schedule Function $\mathcal{G}$	Score↑	Avg Latency (ms)↓
DepthVAR ($\mathcal{R}$=7)	sigmoid@k=12	0.7256	1168
	sigmoid@k=3	0.7175	1139
	sigmoid@k=256	0.7212	1199
	linear-a	0.7218	1177
	linear-b	0.7250	1120
DepthVAR ($\mathcal{R}$=8)	sigmoid@k=12	0.7262	1295
	sigmoid@k=3	0.7200	1285
	sigmoid@k=256	0.7268	1314
	linear-a	0.7228	1316
	linear-b	0.7233	1284
DepthVAR ($\mathcal{R}$=9)	sigmoid@k=12	0.7318	1622
	sigmoid@k=3	0.7316	1576
	sigmoid@k=256	0.7310	1664
	linear-a	0.7310	1573
	linear-b	0.7291	1564

Choice of Schedule Functions. The choice of the schedule function $\mathcal{G}$ determines how computational depth is distributed across tokens. To investigate the impact of different allocation strategies, we experiment with several functional forms for $\mathcal{G}$. As illustrated in Figure 8, these functions create distinct profiles for assigning layer depths based on to-

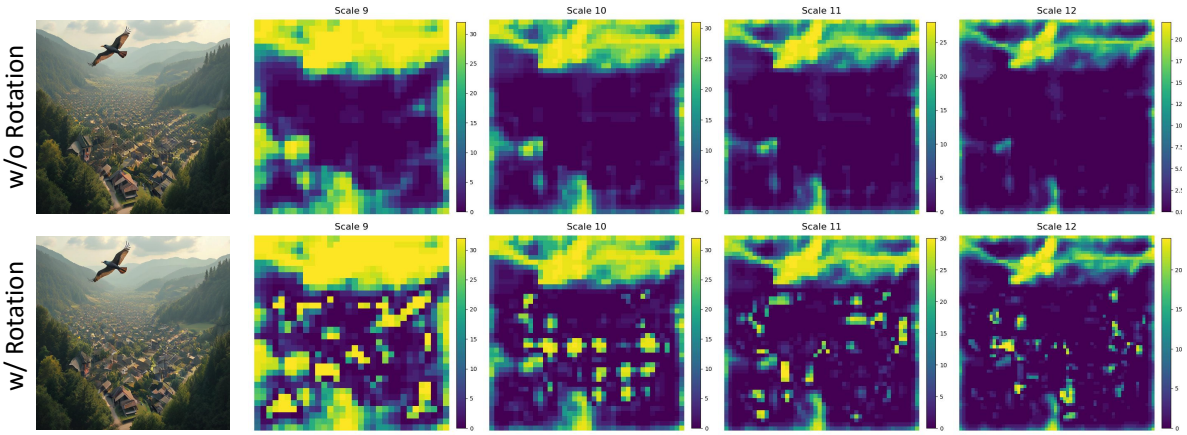


Figure 7. Visualization of depth maps in the presence and absence of Cyclic Percentile Rotation. This rotation operation enables updates in low-score regions that would otherwise remain unchanged. We present the visualizations of the last 4 scales (9-12) for clarity.

Table 4. Ablation study on the impact of Depth-based Code Blending. Disabling it degrades performance on GenEval.

Methods	GenEval	
	Score↑	Latency (ms)↓
DepthVAR($\mathcal{R}$=7)	0.7256	1168
w/o Blend.	0.7102	1172
DepthVAR($\mathcal{R}$=8)	0.7262	1295
w/o Blend.	0.7242	1294
DepthVAR($\mathcal{R}$=9)	0.7318	1622
w/o Blend.	0.7241	1625

Table 5. Ablation study on Cyclic Percentile Rotation. The table reports results with and without this component, illustrating its impact on overall performance.

Methods	GenEval		ImageReward	
	Score↑	Latency (ms)↓	Score↑	Latency (ms)↓
DepthVAR($\mathcal{R}$=7)	0.7256	1168	0.9088	1174
w/o Rotation	0.7277	1088	0.9031	1096
DepthVAR($\mathcal{R}$=8)	0.7262	1295	0.9171	1303
w/o Rotation	0.7222	1206	0.9133	1231
DepthVAR($\mathcal{R}$=9)	0.7318	1622	0.9254	1616
w/o Rotation	0.7274	1545	0.9231	1546

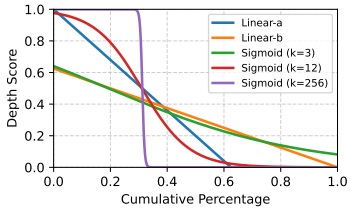


Figure 8. Profiles of different schedule functions under the same constraint. Linear a and b pass $(0, 1)$ and $(1, 0)$.

ken rank, ranging from gradual tapering (sigmoid) to linear decay. We evaluate their effectiveness by comparing GenEval scores across different compute constraints (Table 3). The sigmoid function ($\mathcal{G}_{\text{sigmoid}}$ with $k = 12$) almost consistently achieves the best performance, validating our approach of assigning varied layer depths across tokens. It also suggests that it is not necessary to enforce that all tokens are processed (as in $\mathcal{G}_{\text{linear-b}}$) or that some tokens always traverse the full network depth ($\mathcal{G}_{\text{linear-a}}$). This validates our continuous compute allocation strategy, demonstrating the effectiveness of a balanced schedule profile.

Impact of Cyclic Percentile Rotation. Cyclic percentile rotation plays a critical role in re-ranking tokens for mapping depth scores, breaking the self-reinforcing top selection pattern that often emerges in hard-pruning. An ablation study in Tab. 5 shows that incorporating cyclic percentile rotation consistently yields quality gains. Moreover, as illustrated in Fig. 7, applying it enables regions with originally low scores in the decision map to receive substantive updates. With compute constraints relative to reference scales, cyclic percentile rotation ensures that tokens across spatial regions are iteratively and more evenly updated, rather than focusing updates only on a small subset.

5. Conclusion

In this paper, we introduce DepthVAR, a training-free framework that enables dynamic and continuous computational allocation for each token in Visual Autoregressive models. We analyze the limitations of frequency-based assumptions in hard-pruning acceleration methods and identify exploitable depth redundancy in VAR models. To leverage this redundancy and overcome the pitfalls of binary pruning, DepthVAR employs an adaptive depth scheduler with a cyclic rotated schedule function to heuristically assign computational depth per token. This is realized through a dynamic inference process using a bit-reversal layer-major mask and depth-based code blending. Experiments validate that DepthVAR achieves a superior trade-off between inference latency and performance on various semantic and human preference benchmarks compared to prior hard-pruning approaches, confirming its effectiveness.

Limitations and Future Work. A limitation of our work is the fixed per-sample compute budget. Future work could explore dynamic total compute allocation via routing or early-exiting, and investigate more advanced strategies for exploiting depth redundancy.

Acknowledgments

This work was supported in part by the Open Fund of the Zhongguancun Open Laboratory of Optoelectronic Measurement and Intelligent Perception under the project 'Lightweight Algorithm Design for Multimodal Object Recognition on Spaceborne Platforms'.

References

[1] Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jin Jin, Xin Jiang, Qun Liu, Michael Lyu, and Irwin King. Binarybert: Pushing the limit of bert quantization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4334–4348, 2021. 2

[2] Tolga Bolukbasi, Joseph Wang, Ofer Dekel, and Venkatesh Saligrama. Adaptive neural networks for efficient inference. In *International conference on machine learning*, pages 527–536. PMLR, 2017. 2

[3] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations*, 2023. 5, 6

[4] Zhekai Chen, Ruihang Chu, Yukang Chen, Shiwei Zhang, Yujie Wei, Yingya Zhang, and Xihui Liu. Tts-var: A test-time scaling framework for visual auto-regressive generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. 2

[5] Zhuokun Chen, Jugang Fan, Zhuowei Yu, Bohan Zhuang, and Mingkui Tan. Frequency-aware autoregressive modeling for efficient high-resolution image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17140–17149, 2025. 1, 3, 4, 5, 6, 7

[6] Zigeng Chen, Xinyin Ma, Gongfan Fang, and Xinchao Wang. Collaborative decoding makes visual auto-regressive modeling efficient. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 23334–23344, 2025. 3

[7] Krishna Teja Chitty-Venkata, Sparsh Mittal, Murali Emani, Venkatram Vishwanath, and Arun K Somani. A survey of techniques for optimizing transformer inference. *Journal of Systems Architecture*, 144:102990, 2023. 2

[8] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965. 4

[9] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022. 6

[10] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. *arXiv preprint arXiv:1807.03819*, 2018. 2

[11] Luciano Del Corro, Allie Del Giorno, Sahaj Agarwal, Bin Yu, Ahmed Awadallah, and Subhabrata Mukherjee. Skipdecode: Autoregressive skip decoding with batching and caching for efficient llm inference. *arXiv preprint arXiv:2307.02628*, 2023. 2

[12] 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

[13] Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli. Depth-adaptive transformer. In *ICLR 2020-Eighth International Conference on Learning Representations*, pages 1–14, 2020. 2, 5

[14] Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, et al. Layerskip: Enabling early exit inference and self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12622–12642, 2024. 2, 3

[15] Anne C Elster. Fast bit-reversal algorithms. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 1099–1102. IEEE, 1989. 4

[16] Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. *arXiv preprint arXiv:1909.11556*, 2019. 2, 3

[17] Siqi Fan, Xuezhi Fang, Xingrun Xing, Peng Han, Shuo Shang, and Yequan Wang. Position-aware depth decay decoding (*d*3): Boosting large language model inference efficiency. *arXiv preprint arXiv:2503.08524*, 2025. 2

[18] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. 2

[19] Zhengcong Fei, Xu Yan, Shuhui Wang, and Qi Tian. Deecap: Dynamic early exiting for efficient image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12216–12226, 2022. 2

[20] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36:52132–52152, 2023. 6, 1, 2

[21] Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016. 2, 5

[22] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First conference on language modeling*, 2024. 3

[23] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*, 2023. 2

[24] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19011–19021, 2025. 1, 3, 5, 6, 7, 2

[25] Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bit-

wise autoregressive modeling for high-resolution image synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15733–15744, 2025. 1, 2, 3, 4, 5, 6, 7

[26] Shwai He, Tao Ge, Guoheng Sun, Bowei Tian, Xiaoyang Wang, and Dong Yu. Router-tuning: A simple and effective approach for dynamic depth. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1925–1938, 2025. 2

[27] Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *European Conference on Computer Vision*, pages 289–305. Springer, 2024. 4

[28] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2

[29] Lu Hou, Zhiqi Huang, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. Dynabert: Dynamic bert with adaptive width and depth. *Advances in Neural Information Processing Systems*, 33:9782–9793, 2020. 2

[30] Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens Van Der Maaten, and Kilian Q Weinberger. Multi-scale dense networks for resource efficient image classification. *arXiv preprint arXiv:1703.09844*, 2017. 2

[31] Yekun Ke, Xiaoyu Li, Yingyu Liang, Zhizhou Sha, Zhenmei Shi, and Zhao Song. On computational limits and provably efficient criteria of visual autoregressive models: A fine-grained complexity analysis. *arXiv preprint arXiv:2501.04377*, 2025. 3

[32] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 1

[33] Hermann Kumbong, Xian Liu, Tsung-Yi Lin, Ming-Yu Liu, Xihui Liu, Ziwei Liu, Daniel Y Fu, Christopher Re, and David W Romero. Hmar: Efficient hierarchical masked autoregressive image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2535–2544, 2025. 3

[34] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020. 2

[35] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR, 2023. 3

[36] 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. 3, 5, 6, 7

[37] Kunjun Li, Zigeng Chen, Cheng-Yen Yang, and Jenq-Neng Hwang. Memory-efficient visual autoregressive modeling with scale-aware kv cache compression. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. 3

[38] Ying Li, Chengfei Lv, and Huan Wang. Freqexit: Enabling early-exit inference for visual autoregressive models via frequency-aware guidance. In *NeurIPS*, 2025. 1, 2, 3

[39] Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: Llm quantization with learned rotations. In *The Thirteenth International Conference on Learning Representations*. 2

[40] Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the association for computational linguistics: ACL 2022*, pages 1864–1874, 2022. 3

[41] OpenAI. gpt-oss-120b gpt-oss-20b model card, 2025. 1

[42] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021. 5

[43] David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*, 2024. 2

[44] Sucheng Ren, Yaodong Yu, Nataniel Ruiz, Feng Wang, Alan Yuille, and Cihang Xie. M-var: Decoupled scale-wise autoregressive modeling for high-quality image generation. *arXiv preprint arXiv:2411.10433*, 2024. 3

[45] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019. 2

[46] Tal Schuster, Adam Fisch, Tommi Jaakkola, and Regina Barzilay. Consistent accelerated inference via confident adaptive transformers. *arXiv preprint arXiv:2104.08803*, 2021. 2

[47] Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. *Advances in Neural Information Processing Systems*, 35:17456–17472, 2022. 2, 3, 5

[48] Roy Schwartz, Gabriel Stanovsky, Swabha Swayamdipta, Jesse Dodge, and Noah A Smith. The right tool for the job: Matching model and instance complexities. *arXiv preprint arXiv:2004.07453*, 2020. 2

[49] Weiqiao Shan, Long Meng, Tong Zheng, Yingfeng Luo, Bei Li, Tong Xiao, Jingbo Zhu, et al. Early exit is a natural capability in transformer-based models: An empirical study on early exit without joint optimization. *arXiv preprint arXiv:2412.01455*, 2024. 2

[50] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017. 2

[51] Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Q-bert: Hessian based ultra low precision quantization of bert.

In *Proceedings of the AAAI conference on artificial intelligence*, pages 8815–8821, 2020. 2

[52] Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. Blockwise parallel decoding for deep autoregressive models. *Advances in Neural Information Processing Systems*, 31, 2018. 3

[53] Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. Mobilebert: a compact task-agnostic bert for resource-limited devices. *arXiv preprint arXiv:2004.02984*, 2020. 2

[54] 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. In *The Thirteenth International Conference on Learning Representations*. 1, 2, 3, 5, 6, 7

[55] Shengkun Tang, Yaqing Wang, Zhenglun Kong, Tianchi Zhang, Yao Li, Caiwen Ding, Yanzhi Wang, Yi Liang, and Dongkuan Xu. You need multiple exiting: Dynamic early exiting for accelerating unified vision language model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10791, 2023. 2

[56] Yehui Tang, Yunhe Wang, Jianyuan Guo, Zhijun Tu, Kai Han, Hailin Hu, and Dacheng Tao. A survey on transformer compression. *arXiv preprint arXiv:2402.05964*, 2024. 2

[57] Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd international conference on pattern recognition (ICPR)*, pages 2464–2469. IEEE, 2016. 2

[58] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024. 1, 2, 3

[59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 3

[60] 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. 6, 7, 2, 3

[61] Rui Xie, Tianchen Zhao, Zhihang Yuan, Rui Wan, Wenxi Gao, Zhenhua Zhu, Xuefei Ning, and Yu Wang. Litevar: Compressing visual autoregressive modelling with efficient attention and quantization. In *Workshop on Machine Learning and Compression, NeurIPS 2024*. 3

[62] Ji Xin, Raphael Tang, Yaoliang Yu, and Jimmy Lin. Berxit: Early exiting for bert with better fine-tuning and extension to regression. In *Proceedings of the 16th conference of the European chapter of the association for computational linguistics: Main Volume*, pages 91–104, 2021. 2

[63] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: learning and evaluating human preferences for text-to-image generation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 15903–15935, 2023. 6, 7, 1, 3

[64] Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. Bert loses patience: Fast and robust inference with early exit. *Advances in Neural Information Processing Systems*, 33:18330–18341, 2020. 2

[65] Wei Zhu. Leebert: Learned early exit for bert with cross-level optimization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2968–2980, 2021. 2