

ARVAR: Accelerating Visual Autoregressive Model via Attention Retrospect

Jiedong Zhuang¹, Lu Lu², Ming Dai³, Jian Chen², Qiang Liu², and Haoji Hu¹ *

¹ Zhejiang University, China

² Alibaba Cloud Computing, China

³ Southeast University, China

{zhuangjiedong, haoji_hu}@zju.edu.cn, {mingdai}@seu.edu.cn,
{l1200214, j.chen, yf.yifeng}@alibaba-inc.com



Fig. 1: Our method ARVAR brings a $3.01\times$ acceleration to the Infinity model yet preserves high-fidelity image generation.

Abstract. Visual autoregressive (VAR) models synthesize images in a coarse-to-fine manner by predicting multi-scale token maps, achieving competitive text-to-image quality with fewer decoding iterations than raster-scan autoregression. However, late-scale refinement dominates inference time because computation grows quadratically with resolution and attention becomes increasingly costly. Prior accelerations either filter tokens heuristically, risking severe artifacts, or rely on extra decision models that require per-model fine-tuning with reduced flexibility. We present the first in-depth study of attention behavior across VAR generation steps and uncover two key properties: (1) attention patterns corresponding layers remain consistent across steps; (2) conditional and unconditional branches progressively converge at later scales. Leveraging these insights, we introduce a novel training-free, flashattention-friendly acceleration recipe **ARVAR** with two primary components: (1) **Attention Retrospect (AR)** trims tokens at high resolution phase by token saliency from early stage; (2) **Branch Halving (BH)** drops one of the conditional/unconditional branches in the late steps. Combined, AR and BH deliver faster and higher-quality autoregressive image generation, achieving $1.56\times$ and $3.01\times$ speedups on HART and Infinity respectively with negligible performance drop. Compared with existing acceleration methods, ARVAR substantially improves image fidelity.

Keywords: Efficient · Attention · Visual Autoregressive Models

* Corresponding author.

1 Introduction

Visual Autoregressive (VAR) modeling reframes image synthesis from next-token to next-scale prediction, generating multi-scale token maps that are progressively refined from coarse structure to fine texture [39]. This coarse-to-fine paradigm reduces the number of decoding iterations compared to raster-scan autoregression [4, 6] and has recently produced competitive or superior text-to-image results relative to diffusion-based generators [4, 12, 31, 32]. Scaled systems such as HART [38] and Infinity [16] demonstrate compelling quality and semantic controllability at 1K resolution by combining VAR with strong tokenizers and autoregressive transformers.

Despite these advances, late-scale refinement remains the dominant bottleneck: at each high-resolution step, the model must process the full token grid, causing a quadratic increase in token count with respect to resolution and a sharp rise in attention computation costs [9, 10, 41]. Empirically, the final scales account for the majority of inference time even when memory-efficient attention is enabled [15], as shown in Fig. 2. To address the high inference cost at high resolutions, two complementary strategies have emerged. SkipVAR [21] employs an extra decision model to skip specific generation steps. While it improves image fidelity, it requires per-model fine-tuning, which is time-consuming, computationally expensive and inflexible. FastVAR [15] introduces a plug-and-play scheme that reduces computation at large scales by forwarding only pivotal tokens. However, it adopts a simple, intuitive token filtering mechanism that misses potentially critical tokens, causing severe image distortion. A natural question arises: *Is there a training-free, more precise token selection mechanism that enables token pruning with superior preservation of image fidelity?*

Inspired by evidences in prior works that attention maps are informative for token pruning [5, 51] and fine-grained analysis [11, 25, 49], we posit that attention-derived token importance may distinctly capture structural and semantic cues in VAR. However, in the high-resolution

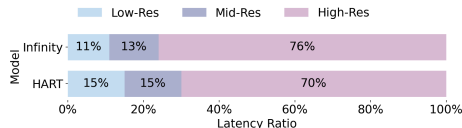


Fig. 2: Inference latency on various steps. The trailing high-resolution steps dominate the runtime budget.

inference phase, producing attention maps online is expensive, as it conflicts with FlashAttention and incurs considerable latency, memory overhead. To overcome these challenges, we first conducting an in-depth study of the attention mechanism across VAR generation steps. Our main observations are as follows:

Insight 1: *Attention patterns (attention maps and outputs) at corresponding layers remain consistent across generation steps.* (Fig. 3) This insight motivates us to leverage cross-step attention consistency to avoid explicitly accessing attention maps during the high-resolution stage; instead, we use attention statistics from the low-resolution stage to drive token selection and output recovery, thereby ensuring compatibility with FlashAttention and maximizing acceleration. Unlike DiT models, this property is unique to VAR: at low-resolution steps it remains highly efficient even without FlashAttention.

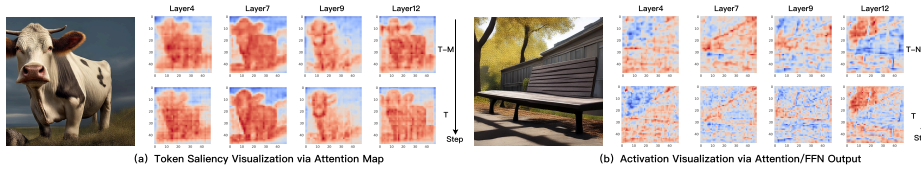


Fig. 3: Attention maps and outputs activation visualization on various layers across distinct generation steps (For clarity, the low-resolution maps are up-sampled to the same resolution as the high-resolution stage). It shows that different steps share similar attention patterns and structurally uniform attention outputs.

Insight 2: *The conditional and unconditional branches gradually converge in later steps.* (Fig. 4) This guides us to safely drop one branch in the final steps, halving computation while preserving image fidelity. By retaining only a single branch’s KV cache, activations, and intermediate results, we significantly reduce peak memory usage.

Based on our observation, we propose a training-free, plug-and-play, high-fidelity acceleration recipe **ARVAR** tailored for VAR models. For token trimming, we introduce **Attention Retrospect (AR)**, which ranks token importance using attention maps from early steps and retains only attentional tokens in subsequent high-resolution steps. Incorporating attention enables more reliable token filtering and finer-grained pruning, thereby improving the preservation of image details. To further accelerate the high-resolution steps, we develop **Branch Halving (BH)** that drops one of the conditional/unconditional branches in the late stage, halving computation with minimal impact on fidelity. Combining these strategies, we deliver faster and higher-quality autoregressive image generation compared to state-of-the-art methods.

In summary, the main contributions of this paper can be outlined as follows:

- We conduct a comprehensive empirical analysis of attention in VAR generation steps, systematically elucidating the attention relationships across different steps.
- We propose a novel, training-free acceleration framework ARVAR, which preserves high image fidelity and delivers $1.56\times$ and $3.01\times$ speedups on HART and Infinity.
- To the best of our knowledge, we are the first to implement attention-driven pruning for VAR models and validate superiority of this route by extensive experiments. A comparison with existing methods is shown in Tab. 1.

2 Related Work

From next-token to next-scale VAR. Early autoregressive image generators adopt the next-token prediction paradigm, treating pixels or quantized patches as discrete tokens

Table 1: Compression scope of VAR model acceleration methods.

Method	High-fidelity	Training-free	Attn-driven
ToMe [1]	✗	✓	✗
FastVAR [15]	✗	✓	✗
SkipVAR [21]	✓	✗	✗
ARVAR	✓	✓	✓

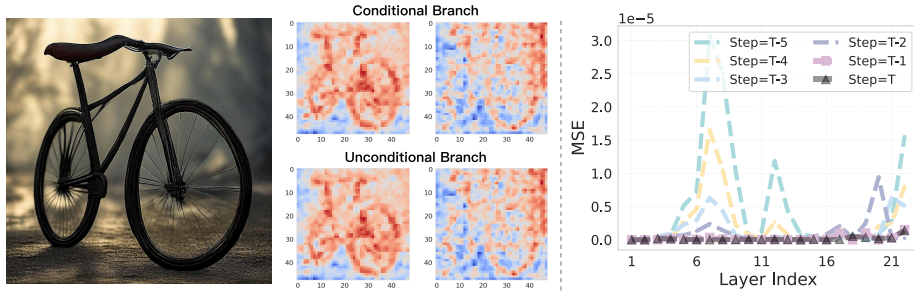


Fig. 4: *Left:* activation visualizations of the conditional and unconditional branches are highly consistent. *Right:* MSE-based consistency check of conditional/unconditional activations over generation steps. Later steps T and $T - 1$ exhibit lower MSE (higher-similarity).

and decoding in raster or sequence order with transformers [6, 13, 30, 36, 40, 44]. While effective, this pipeline suffers from prohibitive decoding steps and substantial attention cost at high resolutions. Visual Autoregressive (VAR) Modeling [39] reframes the unit of autoregression from single tokens to multi-scale token grids and performs next-scale prediction in a coarse-to-fine manner: early small scales capture layout and structure, later large scales refine high-frequency textures. Building on this paradigm, HART [38] and Infinity [16] scale vocabulary, quantization, and bitwise self-correction to deliver high-quality text-to-image synthesis at 1024^2 with competitive efficiency. Nevertheless, the token count and attention cost at late scales still grow rapidly with resolution, making them the dominant latency bottleneck.

Acceleration in visual generation: diffusion versus autoregressive. The diffusion literature offers a rich toolbox for acceleration via distillation [29, 33], quantization [23, 35], token/feature pruning or merging [1, 2, 22, 37, 42], cache technique [26–28, 34, 48, 50, 52–54] and IO-aware attention/backbone engineering [10]. However, direct transfer to VAR is often suboptimal due to the hierarchical, discretized, and multi-scale nature of VAR generation, as well as incompatibilities with KV-caching and the need to preserve spatial lattice structure across scales. In text autoregressive, parallel/speculative decoding [3, 19] reduces steps for next-token prediction, yet it does not immediately apply to VAR where each step processes a token map rather than a single token. These observations motivate VAR-specific accelerators that respect the coarse-to-fine semantics and cross-scale dependencies inherent to next-scale prediction.

VAR-specific accelerators. Recent work focuses on the redundancy and compressibility of late VAR scales. SparseVAR [24] prunes low-frequency tokens and FastVAR [15] identifies that most latency arises at large-scale steps where low-frequency structure has largely converged, and proposes forwarding only high-frequency tokens. In contrast, SkipVAR [21] highlights instance-level heterogeneity in high-frequency sensitivity and employs lightweight decision models over handcrafted frequency features to adaptively select acceleration strategies

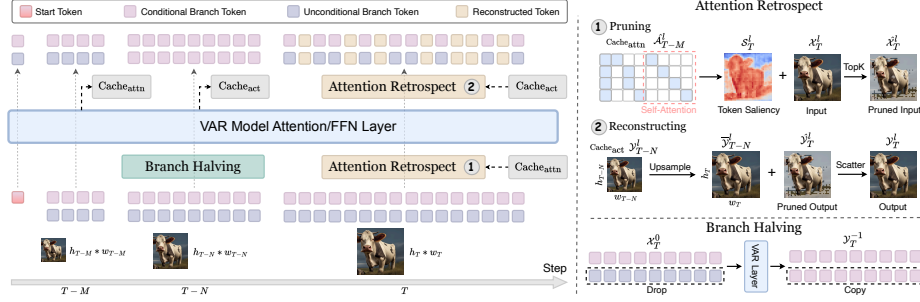


Fig. 5: Pipeline of our ARVAR. We deploy Attention Retrospect (AR) and Branch Halving (BH) at the high-resolution stage to maximize the speedup. AR conducts token pruning by leveraging token saliency computed from low-resolution attention maps and reconstructs the pruned activations via attention outputs from the low-resolution stage. BH skips feeding the unconditional branch and mirrors the conditional output after the layer forward.

per sample. Beyond these post-training methods, LiteVAR [45] explores efficient attention and quantization for structural compression of VAR, and collaborative/ensemble decoding [8] investigates using smaller models or branches to reduce cost at late scales. LANTERN [18] employs speculative decoding to accelerate VAR models. Token-merging approaches [1, 2], while effective in diffusion and ViT inference, typically compress entire token maps and may underperform when naively applied to VAR due to the need to preserve multi-scale spatial correspondences and late-scale texture fidelity.

3 Method

3.1 Preliminary

VAR model. We first summarize the next-scale VAR framework and its standard inference mechanics. Instead of next-token prediction, VAR constructs a hierarchy of token grids $R = \{r_1, \dots, r_t\}$ with a predefined scale schedule (h_t, w_t) increasing toward the target resolution [39]. The joint distribution factorizes autoregressively over scales:

$$p(R) = \prod_{t=1}^T p(r_t | r_{<t}) \quad (1)$$

where text conditioning (for text-to-image) is injected at the earliest step and propagated through the hierarchy [16, 38].

A common practice is to predict residuals relative to an up-sampled intermediate. Let $\tilde{r}_t$ be the intermediate prediction and f_t the transformer output at step t . With an up-sampling operator $\mathbf{Up}(\cdot; h_t, w_t)$,

$$\tilde{r}_t = \mathbf{Up}(\tilde{r}_{t-1}; h_t, w_t) + f_t \quad (2)$$

which unfolds to the cumulative form,

$$\tilde{r}_t = \sum_{i=1}^t \mathbf{Up}(f_i; h_t, w_t) \quad (3)$$

The final token grid $\tilde{r}_t$ is decoded to an image. Empirically, early scales establish low-frequency structure, while late scales refine high-frequency details [15, 21, 39].

At each scale step, the transformer [41] processes the full $h_t \times w_t$ token map (not a single token), making late scales the runtime and memory bottleneck. Attention and feed-forward costs grow super-linearly with the number of tokens $h_t \times w_t$, and overall VAR inference exhibits a steep increase with resolution (often observed as $O(n^4)$ attention for $n \times n$ images) [15, 39]. KV-caching reuses keys/values across scales to avoid re-computation, and IO-aware kernels (e.g., FlashAttention) are compatible with VAR’s block-causal layout [10], though the largest scales still dominate the cost.

Overview. Fig. 5 illustrates the pipeline of our method ARVAR, offering a brief overview of how Attention Retrospect (AR) and Branch Halving (BH) are implemented. We elaborate on AR and BH in the following sections.

3.2 Attention Retrospect

Motivation and Formulation. Visual Auto-Regressive (VAR) models suffer from quadratic computational complexity as token sequences lengthen at higher resolutions. While token pruning is a promising solution, existing attention-based pruning methods are often incompatible with *FlashAttention*, as the latter avoids materializing the full attention matrix to optimize memory. To circumvent this, we propose **Attention Retrospect**, a novel token slimming strategy that leverages historical attention maps as priors for current-step pruning. Standard attention at layer l and step t is formulated as:

$$Q = \mathcal{X}_t^l W_q, \quad K = \mathcal{X}_t^l W_k, \quad V = \mathcal{X}_t^l W_v \quad (4)$$

$$\mathcal{A}_t^l = \text{Softmax}\left(\frac{QK^\top}{\sqrt{D}}\right), \quad \mathcal{Y}_t^l = \text{FFN}(\mathcal{A}_t^l V) \quad (5)$$

where $\mathcal{X}_t^l$ and $\mathcal{Y}_t^l$ denote the input and output features of the l -th attention layer in t -th step, respectively. In the following, T denotes the current high-resolution step.

Attention-driven Pruning. Our empirical observations in Fig. 3(a) suggest that token saliency exhibits significant spatio-temporal consistency across successive VAR steps. Exploiting this, we utilize the attention map $\mathcal{A}_{T-M}^l$ from a preceding lower-resolution step ($T - M$) to guide pruning at step T . Since lower-resolution steps involve fewer tokens, computing the full attention matrix $\mathcal{A}_{T-M}^l$ incurs negligible overhead. We extract the self-attention component, denoted as $\hat{\mathcal{A}}_{T-M}^l$ (Cache_{attn}), which has been shown to better capture intrinsic

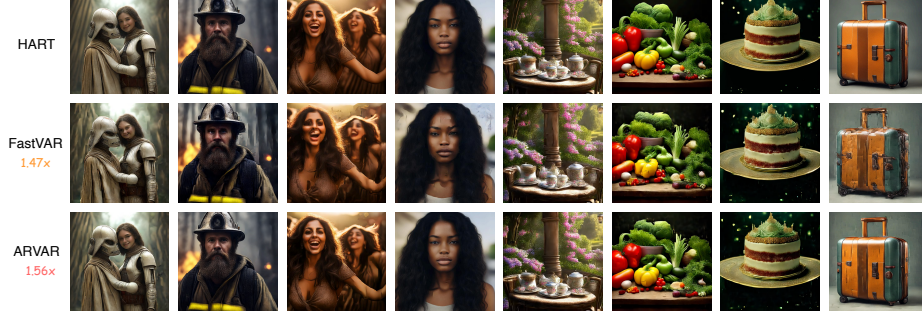


Fig. 6: Comparison between the original model, FastVAR [15] and ARVAR on image generation with HART. Our ARVAR method achieves a higher speedup while delivering better image fidelity compared to FastVAR [15].

image structures [25]. The token importance score $\mathcal{S}_{T-M}^l$ is derived by averaging $\hat{\mathcal{A}}_{T-M}^l$ across columns:

$$\mathcal{S}_{T-M}^l = \frac{1}{C_{T-M}} \sum_{j=1}^{C_{T-M}} \hat{\mathcal{A}}_{T-M}^l[:, j] \quad (6)$$

where $C_{T-M} = h_{T-M} \times w_{T-M}$ represents the sequence length at step $T - M$. To align with the current resolution, we compute the importance score $\mathcal{S}_T^l$ via upsampling: $\mathcal{S}_T^l = \text{Up}(\mathcal{S}_{T-M}^l; h_T, w_T)$. The pruned input $\hat{\mathcal{X}}_T^l$ is then obtained by selecting the top- K most salient tokens:

$$\mathcal{I}_T^l = \text{TopK}(\mathcal{S}_T^l), \quad \hat{\mathcal{X}}_T^l = \text{Pr}(\mathcal{X}_T^l, \mathcal{I}_T^l) \quad (7)$$

where $\text{Pr}(\cdot)$ denotes the pruning operation based on indices $\mathcal{I}_T^l$. Unlike existing intra-layer pruning methods [5, 42], our *pre-step* retrospect mechanism utilizes historical priors to bypass the limitations of *FlashAttention* in high-resolution stages.

Structural Reconstruction. To maintain the 2D grid structure required for subsequent generative layers, we reconstruct the full-resolution activation map. As shown in Fig. 3(b), attention outputs exhibit structural uniformity across steps. We utilize the cached activation $\mathcal{Y}_{T-N}^l$ ($\text{Cache}_{\text{act}}$) to fill the non-salient regions:

$$\bar{\mathcal{Y}}_{T-N}^l = \text{Up}(\mathcal{Y}_{T-N}^l; h_T, w_T) \quad (8)$$

$$\mathcal{Y}_T^l = \text{Sc}(\bar{\mathcal{Y}}_{T-N}^l, \mathcal{I}_T^l, \hat{\mathcal{Y}}_T^l) \quad (9)$$

where $\text{Sc}(\cdot)$ is a scatter operation that updates the upsampled feature map $\bar{\mathcal{Y}}_{T-N}^l$ with the sparse output $\hat{\mathcal{Y}}_T^l$ at indices $\mathcal{I}_T^l$. This *prune-then-reconstruct* paradigm significantly alleviates the computational bottleneck of the attention mechanism while preserving the fidelity of the generated images.

Table 2: Quantitative comparison on GenEval and MJHQ30K benchmarks with SSIM, LPIPS and PSNR image fidelity metrics. The efficiency of HART and Infinity baselines are implemented with FlashAttention.

Methods	Efficiency		GenEval				MJHQ30K				
	Speedup↑	Latency↓	SSIM↑	LPIPS↓	PSNR↑	Acc.↑	SSIM↑	LPIPS↓	PSNR↑	FID↓	CLIP↑
HART [38]	-	0.94s	-	-	-	0.490	-	-	-	30.75	28.51
+FastVAR [15]	1.47×	0.64s	0.6970	0.1597	20.96	0.502	0.6689	0.1434	21.67	28.19	28.39
+ARVAR (Ours)	1.56×	0.60s	0.7455	0.1186	22.47	0.509	0.6904	0.1204	22.00	27.67	28.41
Infinity-2B [16]	-	2.63s	-	-	-	0.677	-	-	-	31.06	28.00
+FastVAR [15]	2.80×	0.94s	0.8198	0.0460	24.71	0.683	0.6870	0.1526	20.47	30.79	27.80
+ARVAR (Ours)	3.01×	0.88s	0.8493	0.0316	25.91	0.685	0.7203	0.1256	22.73	29.65	27.91

3.3 Branch Halving

VAR models employ unconditional branch to provides a data prior that, combined with the conditional branch, enables a controllable trade-off between fidelity. This design captures general structural and temporal priors that complement the conditional semantics, improving global coherence and alignment. However, in our empirical study Fig. 4, we find the two branches typically converge and their activations disparity becomes negligible, leaving the unconditional branch largely superfluous in last step. Building on this observation, we introduce a branch-halving scheme wherein, the unconditional branch is dropped and substituted by the conditional branch in the late generation phase. Consider step T to be branch-redundant, the procedure executed in this step can be formulated as follows:

$$\mathcal{X}_T^0 = [x_{T,\text{cond}}^0; \underbrace{x_{T,\text{uncond}}^0}_{\text{Drop}}] \Rightarrow [x_{T,\text{cond}}^0] \quad (10)$$

$$\mathcal{Y}_T^{-1} = [y_{T,\text{cond}}^{-1}; y_{T,\text{uncond}}^{-1}] \Rightarrow [y_{T,\text{cond}}^{-1}; \underbrace{y_{T,\text{cond}}^{-1}}_{\text{Repeat}}] \quad (11)$$

where $\mathcal{X}_T^0$, $\mathcal{Y}_T^{-1}$ means input and output of all transformer blocks in a step. $x_{T,\text{cond}}^0$, $x_{T,\text{uncond}}^0$ denotes input of conditional/unconditional branch, which are concatenated along the batch dimension. Consequently, our method delivers a 50% FLOPs reduction at step T , decreasing wall-clock inference time. This technique is fully orthogonal to token pruning routes; it can be combined with them and remains fully compatible with FlashAttention.

4 Experiments

4.1 Implementation and evaluation metrics

We integrate ARVAR into two representative VAR-based text-to-image backbones HART [38] and Infinite [16], both capable of synthesizing images at 1024²

Table 3: Quantitative comparison on efficiency and quality on 1024×1024 GenEval benchmarks. The marks $\bullet$, $\circ$, and $\diamond$ denote the Diffusion, Autoregressive, and VAR-based methods, respectively.

Methods	Inference Efficiency			GenEval Metric						
	Steps↓	Speedup↑	Latency↓	colors↑	two_object↑	one_object↑	position↑	color_attr↑	counting↑	Overall↑
$\bullet$ SDXL [31]	40	-	4.4s	0.83	0.72	0.97	0.16	0.24	0.35	0.55
$\bullet$ PixArt-Sigma [4]	20	-	2.6s	0.85	0.60	0.99	0.15	0.28	0.39	0.54
$\bullet$ SD3-medium [12]	28	-	4.5s	0.85	0.72	0.97	0.35	0.37	0.37	0.62
$\circ$ LlamaGen [36]	1024	-	40.2s	0.60	0.35	0.70	0.07	0.05	0.15	0.32
$\circ$ Show-o [44]	1024	-	48.7s	0.89	0.80	0.99	0.31	0.50	0.66	0.65
$\circ$ HART [38]	14	-	0.94s	0.79	0.59	0.95	0.11	0.18	0.31	0.49
$\diamond$ +FastVAR [15]	14	1.47×	0.64s	0.81	0.58	0.91	0.14	0.18	0.36	0.50
$\diamond$ +ARVAR (Ours)	14	1.56×	0.60s	0.83	0.61	0.93	0.14	0.19	0.36	0.51
$\diamond$ Infinity-2B [16]	13	-	2.63s	0.85	0.78	0.99	0.25	0.57	0.61	0.68
$\diamond$ +FastVAR [15]	13	2.80×	0.94s	0.82	0.79	0.99	0.27	0.60	0.64	0.68
$\diamond$ +ARVAR (Ours)	13	3.01×	0.88s	0.85	0.78	0.99	0.27	0.61	0.63	0.69

resolution. To ensure an apples-to-apples comparison, we leave all hyperparameters at the default settings of baselines [1, 15]. Our evaluation spans two axes: generation quality and inference efficiency. For quality, we report GenEval [14] to assess high-level semantic alignment and MJHQ30K [20] to gauge perceptual fidelity. Distinct from prior work [15], we augment evaluation with SSIM [17], LPIPS [47] and PSNR [17] to assess the optimized model versus the original, which are more indicative of image fidelity. For efficiency, we measure end-to-end runtime, speedup over the baseline, and peak GPU memory footprint. Unless specified, all reported efficiency results were collected with FlashAttention-enabled. All experiments are conducted on one NVIDIA RTX3090 GPU.

4.2 Main results

Comparison on GenEval and MJHQ30K. We evaluate on GenEval [14] and MJHQ30K [20] with SSIM, LPIPS, PSNR (fidelity), Acc. (GenEval), FID/CLIP (MJHQ30K), and report efficiency via speedup and latency, using FlashAttention across baselines in Tab. 2. Across both HART and Infinity backbones, ARVAR consistently delivers a improvement over FastVAR with higher speedup and lower latency while improving image fidelity. On HART, ARVAR achieves $1.56\times$ speedup and reduces latency to 0.60s, while increasing SSIM/PSNR/Acc./CLIP score and reducing LPIPS/FID on GenEval and MJHQ30K. On Infinity, ARVAR reaches $3.01\times$ speedup and lowers latency 0.88s, while consistently improving above image quality metrics. Overall, with FlashAttention enabled, ARVAR offers a superior speed-quality trade-off and serves as a strong drop-in accelerator for existing backbones. A qualitative visualization result are presented in Fig. 6.

Comparison with Diffusion Model. We benchmark diffusion, autoregressive, and VAR families on the 1024×1024 GenEval suite, reporting steps, speedup, latency and per-category success rates plus the Overall score in Tab. 3. VAR methods operate with 13–14 steps and substantially lower latency than diffusion (20–40 steps, 2.6–4.5s) and autoregressive models (1024 steps, 40–48s). Among baselines, Infinity yields the strongest speed-quality trade-off (0.68 Overall at

Table 4: Comparison with ToMe [1] and SparseVAR [7].

Method	Speedup $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	FID $\downarrow$	CLIP $\uparrow$
HART	1.00 $\times$	-	-	-	30.75	28.51
ToMe [1]	1.40 $\times$	0.6047	0.1893	17.69	29.88	28.23
Sparse [7]	1.50 $\times$	0.6425	0.1677	19.21	28.33	28.20
ARVAR	1.56$\times$	0.6904	0.1204	22.00	27.67	28.41

Table 5: Ablation on AR and BH on MJHQ30K with HART.

	AR	BH	Speedup $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	FID $\downarrow$	CLIP $\uparrow$
			1.00 $\times$	-	-	-	30.75	28.51
$\checkmark$			1.46 $\times$	0.7232	0.1088	22.59	27.56	28.50
	$\checkmark$		1.43 $\times$	0.7567	0.0523	23.90	28.01	28.35
$\checkmark$	$\checkmark$		1.56 $\times$	0.6904	0.1204	22.00	27.67	28.41

2.63s). Building on these backbones, our ARVAR consistently advances the frontier over FastVAR and the vanilla models.

Comparison with other baselines. In Tab. 4, we compare our method to the token-merging baseline ToMe [1] and pruning baseline SparseVAR [24] on the HART backbone. ARVAR achieves the highest efficiency and quality, delivering 1.56 $\times$ speedup while improving all fidelity metrics. ARVAR also achieves a higher CLIP score 28.41 than ToMe and LiteVAR. Relative to the HART baseline, ARVAR provides a substantial acceleration 1.56 $\times$ and the largest FID reduction (3.08, about 10%), with a negligible CLIP change. The results demonstrate the ARVAR is superior to these techniques, offering higher speedup without the pronounced perceptual degradation.

Comparison on HPSv2.1 and ImageReward. We test our method on other benchmarks HPSv2.1 [43] and ImageReward [46] in Tab. 6, comparing with ToMe [1] and FastVAR [15] together with speed and latency. ARVAR delivers the highest speedup and lowest latency in both backbones with highest image quality metrics. It offers the most favorable efficiency-performance balance, demonstrating that attention-driven token pruning scales well to larger backbones and remains practical for deployment.

4.3 Ablations

Attention Retrospect and Branch Halving. We ablate Attention Retrospect and Branch Halving on MJHQ30K to disentangle their efficiency and quality contributions in Tab. 5. Individually, both modules accelerate inference and improve perceptual quality over the baseline: AR yields 1.46 $\times$ speedup with the strongest FID gain 27.56 and near-

unchanged CLIP score, while BH provides 1.43 $\times$ speedup and the best pixel-level fidelity. Combining AR+BH maximizes efficiency 1.56 $\times$ and retains a strong perceptual score 27.67 with a small CLIP score degradation. Notably, with a comparable speedup ratio (1.46 $\times$ vs 1.47 $\times$ in Tab. 2), AR achieves significantly superior image quality over FastVAR [15]. In summary, AR+BH offers the best speed while preserving most of the perceptual improvement, providing a tunable trade-off tailored to the priorities of speed and perceptual quality.

Table 6: Comparison on HPSv2.1, ImageReward benchmarks.

Method	Speedup $\uparrow$	Latency $\downarrow$	HSPv2.1 $\uparrow$	ImageReward $\uparrow$
HART	1.00 $\times$	0.94s	28.72	0.5634
ToMe [1]	1.40 $\times$	0.67s	27.32	0.4973
FastVAR [15]	1.47 $\times$	0.64s	27.92	0.5329
ARVAR	1.56$\times$	0.60s	27.67	0.5511
Infinity-2B	1.00 $\times$	2.63s	30.12	0.9199
ToMe [1]	2.67 $\times$	0.99s	29.74	0.8875
FastVAR [15]	2.80 $\times$	0.94s	29.92	0.9046
ARVAR	3.01$\times$	0.88s	30.05	0.9118

Table 7: Ablation on token pruning metric of Attention Retrospect on MJHQ30K benchmark with HART. “Full-attn” and “Self-attn” indicates full attention and self-attention for token saliency computation.

Methods	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	FID $\downarrow$	CLIP $\uparrow$
HART	-	-	-	30.75	28.51
Random	0.4537	0.3821	15.23	31.98	27.23
Frequency [15]	0.6689	0.1434	21.67	28.19	28.39
AR (Full-attn)	0.6734	0.1388	21.59	28.23	28.45
AR (Self-attn)	0.6904	0.1204	22.00	27.67	28.41

Table 8: Ablation experiments on token pruning step/ratio and branch drop, where “ St ” denotes the t -th step and “BH” means Branch Halving. We label pruning at steps 13 as {S13}.

T	M	N	Ratio	BH	Speedup $\uparrow$	FID $\downarrow$
Baseline	-	-	-	-	1.00 $\times$	30.75
{S13}	3	1	50%	{S14}	1.43 $\times$	28.12
{S13}	2	1	50%	{S14}	1.39 $\times$	27.85
{S14}	3	2	75%	{S13;S14}	1.60$\times$	27.73
{S14}	3	2	75%	{S13}	1.56 $\times$	27.67

Token pruning metric. We exam the token-pruning criterion in Attention Retrospect, comparing Random, Frequency (FastVAR) and saliency from full/self-attention. Random pruning severely degrades quality, highlighting the need for saliency-aware pruning. Frequency performs substantially better, but AR-based saliency dominates it: Self-attn achieves the best overall fidelity and perceptual quality improving over Frequency by +0.0215 SSIM, -0.0230 LPIPS, +0.33 PSNR and -0.52 FID with essentially robust CLIP. Full-attn is close and yields the highest CLIP, but lags Self-attn on distortion metrics and FID, suggesting self-attention provides more discriminative token importance for pruning. We qualitatively analyze the token selected by self-attention saliency in Fig. 9.

Hyperparameters examination. An ablation on the token-pruning schedule (step and ratio) and the placement of Branch Halving on MJHQ30K is shown in Tab. 8. Pruning later in the high-resolution phase is clearly preferable: moving from S13 (50% pruning) to S14 (75% pruning) improves both efficiency and perceptual quality. At S13, adopting later-stage attention maps slightly lowers speedup but improves FID from 28.12 to 27.85, indicating that later attention maps offer superior token screening, albeit with a small increase in overhead. The best FID is achieved by pruning at S14 with a 75% ratio and applying BH at a single earlier step S13, reaching 27.67 FID with 1.56 $\times$ speedup. These results suggest a practical recipe: prune later more aggressively and place BH at a single step to maximize perceptual quality at near-maximal speed; add BH at both steps when peak throughput is prioritized with minimal quality loss.

Trade off. Fig. 7 plots the speed–quality trade-off for HART and Infinity under increasing acceleration. As expected, quality decreases with higher speedup for both methods. However, ARVAR consistently dominates FastVAR across all operating points and both backbones, achieving higher SSIM and PSNR at matched speedups. The margin grows at more aggressive acceleration, indicating ARVAR’s greater robustness to compression. On HART, ARVAR maintains noticeably higher SSIM/PSNR even at the largest tested speedup, whereas FastVAR degrades more steeply. On Infinity, ARVAR shifts the Pareto frontier upward, reaching comparable or better quality than FastVAR at clearly higher speedups. The curves demonstrate that ARVAR provides a superior and more stable speed–quality trade-off, especially in the high-acceleration regime. Fig. 8

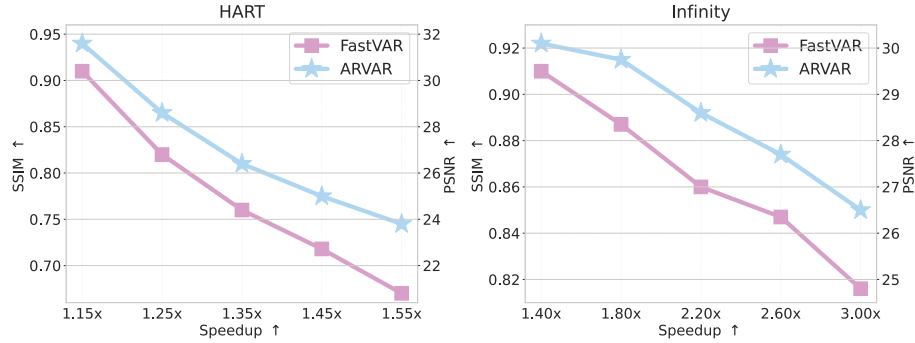


Fig. 7: The trade-off between acceleration and image fidelity for HART and INfinity on the Geneval [14] dataset.



Fig. 8: Generated images under various speedup with ARVAR.

offers a visualization of images generated by HART with ARVAR. Even at the maximum speedup, visual quality remains intact.

4.4 Analysis

Qualitative and quantitative analysis of Token Selection. Fig. 9 analyzes token selection qualitatively. ARVAR produces spatially coherent pruning masks that concentrate on background regions and preserve foreground objects, whereas FastVAR exhibits scattered, fragmented drops that cut through salient structures. In Fig. 10, we analyze token selection quantitatively and check the top-k important-token overlap between an intermediate step t and the final step T . It reveals a robustness regime around $T - 3$: overlap reaches roughly 94-95% (top-60%), 92-93% (top-50%), and 84-85% (top-40%), and further climbs to about 97%, 96%, and 89% at $T - 1$. These trends indicate that early-nearby stage attention yields more reliable saliency for pruning, enabling finer-grained control and higher precision in token selection. Guided by these observations, ARVAR leverages early-nearby step attention to produce coherent masks that align with object boundaries, which in turn supports the superior speed-quality trade-off reported in our main results.

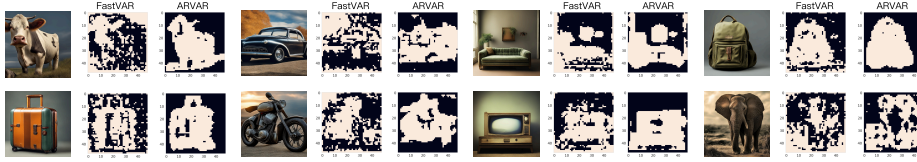


Fig. 9: Token selection visualization, with black marking dropped tokens. Our ARVAR provides finer-grained control and higher precision.

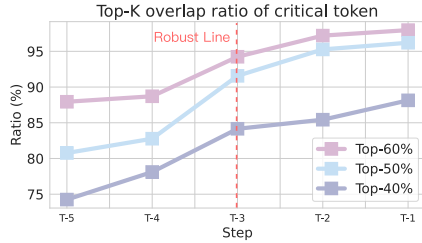


Fig. 10: The top-k important-token overlap (Step t vs. Step T) increases as t approaches T .

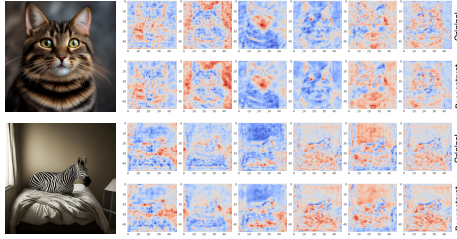


Fig. 11: Visualization of the Attention+FFN outputs from different layers.

Qualitative analysis of reconstruction. Fig. 11 provides a layer-wise qualitative check of feature fidelity. We visualize the Attention+FFN outputs at several depths and compare the original feature maps with those reconstructed by our method. Across layers, the reconstructed maps preserve the spatial layout and activation polarity of salient regions, maintaining coherent object boundaries and background suppression. Differences are minor and mostly localized to fine textures, while large-scale structures remain unchanged, indicating low information loss through our approximation. This consistency from shallow to deep layers supports the faithfulness of our reconstruction mechanism and helps explain the strong perceptual quality observed at high acceleration.

Visualization of Distinct Steps. Fig. 12 visualizes the intermediate outputs at different steps. This finding supports our observation that the stepwise stability of attention patterns stems from the inherent consistency of image structure, verifying the efficacy of our attention retrospect scheme.

Efficiency Check. Fig. 13 shows the latency on different modules in a transformer block of HART with or without ARVAR. ARVAR cuts Attention/FFN from 11.23ms to 2.33ms ($4.8\times$) and 5.24ms to 1.30ms ($4.0\times$), while introduce small overhead that 0.55ms for pruning and 0.21ms for reconstruction (0.76ms in total). Overall, our ARVAR eliminates the attention bottleneck and substantially reduces FFN cost, while introducing only minor overhead.

Memory footprint analysis. We evaluate memory and latency under the FlashAttention setting. Naive scaled dot-product attention runs out of memory, underscoring the need for memory-efficient kernels at this resolution. With FlashAttention, the baseline fits in 21.45 GB and runs at 0.94s. Adding our ARVAR yields a $1.56\times$ speedup with latency reduced to 0.60s while keeping

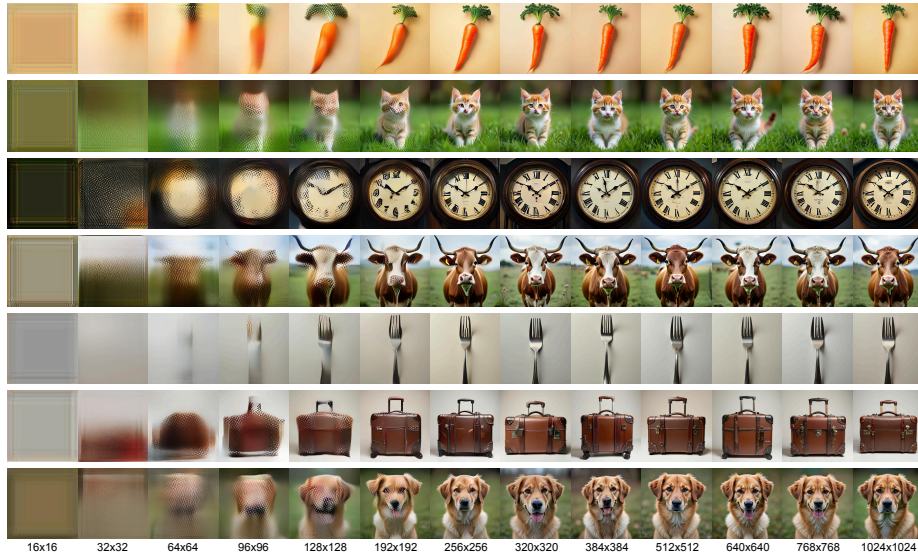


Fig. 12: Visualizations of the intermediate outputs at different steps for Infinity [16].

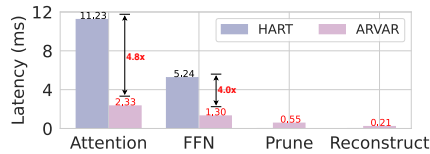


Fig. 13: Comparison on efficiency of Attention and FFN.

Table 9: Memory and latency examination with FlashAttention.

Method	Speedup↑	Latency↓	GPU Memory ↓
NaiveAttn [41]	-	-	OOM
FlashAttn [9]	1.00×	0.94s	21.45GB
FlashAttn+ARVAR	1.56×	0.60s	21.33GB
—Cache _{attn}	-	0.007s 1.2%	0.05GB 0.2%
—Cache _{act}	-	-	0.18GB 0.8%

the memory footprint essentially unchanged at 21.33 GB. The overhead introduced by $\text{Cache}_{\text{attn}}$ and $\text{Cache}_{\text{act}}$ for AR is around 1% or less of the overall latency and memory footprint. These results show that ARVAR is lightweight and complementary to FlashAttention: it delivers substantial compute savings without introducing memory overhead, making it a practical drop-in accelerator for high-resolution inference.

Failure cases. As shown in Fig. 14, BH failures mostly follow three patterns: (i) *fine binding collapse*, where coarse semantics are preserved but precise attribute/object assignment weakens; (ii) *rare-attribute collapse*, where unusual text-specified properties drift toward more natural visual priors; and (iii) *asymmetry collapse*, where subtle identity-sensitive or locally asymmetric details become more generic or symmetric. Removing the unconditional branch may weaken the late-stage classifier-free guidance needed to maintain fine semantic disambiguation, unusual attributes, and asymmetric details.

Pseudo code. In Algo. 1, we present the overall pipeline of the Pytorch-Implemented pseudocode for Attention Retrospect. Our method is plug-and-play, it can be seamlessly integrated into various VAR models using a few code lines with negligible overhead.

Algorithm 1: Attention Retrospect Pipeline

Input : Attention cache $\mathbf{C}_{\text{attn}}$, Activation cache $\mathbf{C}_{\text{act}}$, Input sequence $\mathbf{X}_{\text{in}} \in \mathbb{R}^{B \times T \times D}$

Output: Output sequence $\mathbf{Y}_{\text{out}} \in \mathbb{R}^{B \times T \times D}$

$h, w \leftarrow \text{int}(\sqrt{T}), \text{int}(\sqrt{T})$ // current spatial size (assume $T = h \times w$)

$\mathbf{A} \leftarrow \text{mean}(\mathbf{C}_{\text{attn}}, -1)$ // average over column: $B \times T$ or $B \times T \times 1$

$\mathbf{A}_{\uparrow} \leftarrow \text{Interp}(\mathbf{A} \rightarrow (h, w)).\text{reshape}(B, h \cdot w)$ // upsample saliency to current size

$\mathbf{I} \leftarrow \text{TopKIndices}(\mathbf{A}_{\uparrow}, k = \text{topk}, \text{descending} = \text{True})$ // important token indices

$\mathbf{X}_{\text{keep}} \leftarrow \text{Gather}(\mathbf{X}_{\text{in}}, \text{dim} = 1, \text{index} = \mathbf{I})$ // token pruning

$\mathbf{Y}_{\text{keep}} \leftarrow \text{TransformerBlock}(\mathbf{X}_{\text{keep}})$ // forward on kept tokens

$\mathbf{V}_{\uparrow} \leftarrow \text{Interp}(\mathbf{C}_{\text{act}} \rightarrow (h, w))$ // upsample activation to current size

$\mathbf{Y}_{\text{out}} \leftarrow \mathbf{V}_{\uparrow}$ // initialize with (upsampled) cached activations

$\mathbf{Y}_{\text{out}} \leftarrow \text{Scatter_}(\mathbf{Y}_{\text{out}}, \text{dim} = 1, \text{index} = \mathbf{I}, \text{src} = \mathbf{Y}_{\text{keep}})$ // reconstruction

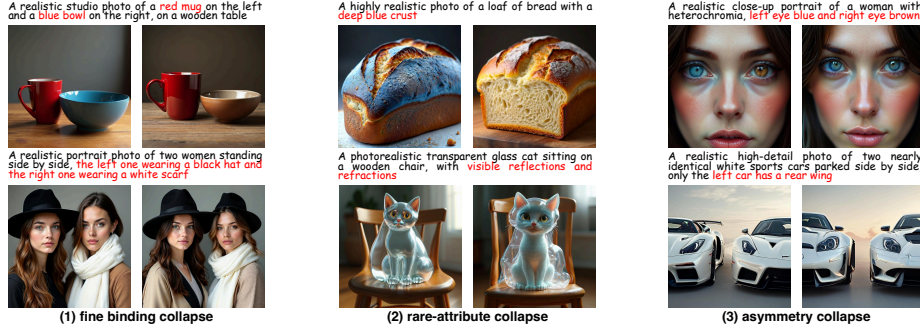


Fig. 14: Representative failure cases of BH.

5 Conclusion

This paper presents a novel acceleration framework **ARVAR** for Visual Autoregressive (VAR) Model. We first dive into attention properties within VAR and discover that (1) attention patterns remain consistent across generation steps; (2) the conditional and unconditional branches gradually converge in later steps. Motivated by these insights, we propose Attention Retrospect and Branch Halving, which significantly overcome the latency bottleneck at VAR’s high-resolution phase. To the best of our knowledge, we are the first to explore attention-driven token pruning in VARs, demonstrate its effectiveness, and show full compatibility with FlashAttention. We envision that this work could usher in new acceleration techniques (e.g. token reduction, redundant structure cutting) and inspire the research community to enhance the understanding of VAR models.

References

1. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your vit but faster. arXiv preprint arXiv:2210.09461 (2022)
2. Bolya, D., Hoffman, J.: Token merging for fast stable diffusion. In: CVPR (2023)
3. Cai, T., Li, Y., Geng, Z., Peng, H., Lee, J.D., Chen, D., Dao, T.: Medusa: Simple llm inference acceleration framework with multiple decoding heads. arXiv preprint arXiv:2401.10774 (2024)
4. Chen, J., Ge, C., Xie, E., Wu, Y., Yao, L., Ren, X., Wang, Z., Luo, P., Lu, H., Li, Z.: Pixart-sigma: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. In: ECCV (2024)
5. Chen, L., Zhao, H., Liu, T., Bai, S., Lin, J., Zhou, C., Chang, B.: An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In: European Conference on Computer Vision. pp. 19–35. Springer (2024)
6. Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., Sutskever, I.: Generative pretraining from pixels. In: ICML (2020)
7. Chen, Z., Fan, J., Yu, Z., Zhuang, B., Tan, M.: Frequency-aware autoregressive modeling for efficient high-resolution image synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17140–17149 (2025)
8. Chen, Z., Ma, X., Fang, G., Wang, X.: Collaborative decoding makes visual autoregressive modeling efficient. arXiv preprint arXiv:2411.17787 (2024)
9. Dao, T.: Flashattention-2: Faster attention with better parallelism and work partitioning. In: ICLR (2024)
10. Dao, T., Fu, D.Y., Ermon, S., Rudra, A., R’e, C.: Flashattention: Fast and memory-efficient exact attention with io-awareness. In: NeurIPS (2022)
11. Darcet, T., Oquab, M., Mairal, J., Bojanowski, P.: Vision transformers need registers. arXiv preprint arXiv:2309.16588 (2023)
12. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Muller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: ICML (2024)
13. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: CVPR (2021)
14. Ghosh, D., Hajishirzi, H., Schmidt, L.: Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems* **36**, 52132–52152 (2023)
15. Guo, H., Li, Y., Zhang, T., Wang, J., Dai, T., Xia, S.T., Benini, L.: Fastvar: Linear visual autoregressive modeling via cached token pruning. arXiv preprint arXiv:2503.23367 (2025)
16. Han, J., Liu, J., Jiang, Y., Yan, B., Zhang, Y., Yuan, Z., Peng, B., Liu, X.: Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. arXiv preprint arXiv:2412.04431 (2024)
17. Hore, A., Ziou, D.: Image quality metrics: Psnr vs. ssim. In: 2010 20th international conference on pattern recognition. pp. 2366–2369. IEEE (2010)
18. Jang, D., Park, S., Yang, J.Y., Jung, Y., Yun, J., Kundu, S., Kim, S.Y., Yang, E.: Lantern: Accelerating visual autoregressive models with relaxed speculative decoding. arXiv preprint arXiv:2410.03355 (2024)
19. Leviathan, Y., Kalman, M., Matias, Y.: Fast inference from transformers via speculative decoding. In: ICML (2023)

20. Li, D., Kamko, A., Akhgari, E., Sabet, A., Xu, L., Doshi, S.: Playground v2.5: Three insights towards enhancing aesthetic quality in text-to-image generation. arXiv preprint arXiv:2402.17245 (2024)
21. Li, J., Ma, Y., Zhang, X., Wei, Q., Liu, S., Zhang, L.: Skipvar: Accelerating visual autoregressive modeling via adaptive frequency-aware skipping. arXiv preprint arXiv:2506.08908 (2025)
22. Li, X., Ma, C., Yang, X., Yang, M.H.: Vidtoime: Video token merging for zero-shot video editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7486–7495 (2024)
23. Li, X., Liu, Y., Lian, L., Yang, H., Dong, Z., Kang, D., Zhang, S., Keutzer, K.: Q-diffusion: Quantizing diffusion models. In: ICCV (2023)
24. Li, Z., Wang, N., Bai, T., Mei, C., Wang, P., Qiu, S., Cheng, J.: SparVAR: Exploring Sparsity in Visual AutoRegressive Modeling for Training-Free Acceleration. arXiv preprint arXiv:2602.04361 (2026)
25. Liu, B., Wang, C., Cao, T., Jia, K., Huang, J.: Towards understanding cross and self-attention in stable diffusion for text-guided image editing. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7817–7826 (2024)
26. Liu, F., Zhang, S., Wang, X., Wei, Y., Qiu, H., Zhao, Y., Zhang, Y., Ye, Q., Wan, F.: Timestep embedding tells: It’s time to cache for video diffusion model. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 7353–7363 (2025)
27. Liu, J., Zou, C., Lyu, Y., Chen, J., Zhang, L.: From reusing to forecasting: Accelerating diffusion models with taylorseers. arXiv preprint arXiv:2503.06923 (2025)
28. Ma, X., Fang, G., Wang, X.: Deepcache: Accelerating diffusion models for free. In: CVPR (2024)
29. Meng, C., Rombach, R., Gao, R., Kingma, D., Ermon, S., Ho, J., Salimans, T.: On distillation of guided diffusion models. In: CVPR (2023)
30. Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A.: Conditional image generation with pixelcnn decoders. In: NeurIPS (2016)
31. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Muller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis. arXiv preprint arXiv:2307.01952 (2023)
32. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022)
33. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. arXiv preprint arXiv:2202.00512 (2022)
34. Selvaraju, P., Ding, T., Chen, T., Zharkov, I., Liang, L.: Fora: Fast-forward caching in diffusion transformer acceleration. arXiv preprint arXiv:2407.01425 (2024)
35. Shang, Y., Yuan, Z., Xie, B., Wu, B., Yan, Y.: Post-training quantization on diffusion models. In: CVPR (2023)
36. Sun, P., Jiang, Y., Chen, S., Zhang, S., Peng, B., Luo, P., Yuan, Z.: Autoregressive model beats diffusion: Llama for scalable image generation. arXiv preprint arXiv:2406.06525 (2024)
37. Sun, W., Tu, R.C., Liao, J., Jin, Z., Tao, D.: Asymrn: Video diffusion transformers acceleration with asymmetric reduction and restoration. arXiv preprint arXiv:2412.11706 (2024)
38. Tang, H., Wu, Y., Yang, S., Xie, E., Chen, J., Chen, J., Zhang, Z., Cai, H., Lu, Y., Han, S.: Hart: Efficient visual generation with hybrid autoregressive transformer. arXiv preprint arXiv:2410.10812 (2024)

39. Tian, K., Jiang, Y., Yuan, Z., Peng, B., Wang, L.: Visual autoregressive modeling: Scalable image generation via next-scale prediction. In: NeurIPS (2024)
40. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. In: NeurIPS (2017)
41. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017)
42. Wang, H., Liu, D., Kang, Y., Li, Y., Lin, Z., Jha, N.K., Liu, Y.: Attention-driven training-free efficiency enhancement of diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16080–16089 (2024)
43. Wu, X., Hao, Y., Sun, K., Chen, Y., Zhu, F., Zhao, R., Li, H.: Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. arXiv preprint arXiv:2306.09341 (2023)
44. Xie, J., Mao, W., Bai, Z., Zhang, D.J., Wang, W., Lin, K.Q., Gu, Y., Chen, Z., Yang, Z., Shou, M.Z.: Show-o: One single transformer to unify multimodal understanding and generation. arXiv preprint arXiv:2408.12528 (2024)
45. Xie, R., Zhao, T., Yuan, Z., Wan, R., Gao, W., Zhu, Z., Ning, X., Wang, Y.: Litevar: Compressing visual autoregressive modelling with efficient attention and quantization. arXiv preprint arXiv:2411.17178 (2024)
46. Xu, J., Liu, X., Wu, Y., Tong, Y., Li, Q., Ding, M., Tang, J., Dong, Y.: Imagereward: Learning and evaluating human preferences for text-to-image generation. Advances in Neural Information Processing Systems **36**, 15903–15935 (2023)
47. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
48. Zhao, X., Jin, X., Wang, K., You, Y.: Real-time video generation with pyramid attention broadcast. arXiv preprint arXiv:2408.12588 (2024)
49. Zhuang, J., Hu, J., Mu, L., Hu, R., Liang, X., Ye, J., Hu, H.: Falip: Visual prompt as foveal attention boosts clip zero-shot performance. In: European Conference on Computer Vision. pp. 236–253. Springer (2024)
50. Zhuang, J., Lu, L., Dai, M., Chen, J., Liu, Q., Hu, H.: Myopia rectification: Kv cache pruning for mllms via dynamic attention subsidy and token reclamation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9023–9033 (2026)
51. Zhuang, J., Lu, L., Dai, M., Hu, R., Chen, J., Liu, Q., Hu, H.: St3: Accelerating multimodal large language model by spatial-temporal visual token trimming. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 11049–11057 (2025)
52. Zhuang, J., Lu, L., Dai, M., Hu, R., Chen, J., Liu, Q., Hu, H.: Q cache: Visual attention is valuable in less than half of decode layers for multimodal large language model. arXiv preprint arXiv:2602.01901 (2026)
53. Zou, C., Liu, X., Liu, T., Huang, S., Zhang, L.: Accelerating diffusion transformers with token-wise feature caching. arXiv preprint arXiv:2410.05317 (2024)
54. Zou, C., Zhang, E., Guo, R., Xu, H., He, C., Hu, X., Zhang, L.: Accelerating diffusion transformers with dual feature caching. arXiv preprint arXiv:2412.18911 (2024)