VibeToken: Scaling 1D Image Tokenizers and Autoregressive Models for Dynamic Resolution Generations

Anonymous CVPR submission

Figure 1. **VibeToken-Gen image generation examples.** A single resolution-generalist visual autoregressive model trained on ImageNet1k that can synthesize arbitrary resolution and aspect ratio images. The examples shown are from various resolutions between 256×256 and 1024×1024. **VibeToken-Gen** leverages our novel *resolution-agnostic* 1D visual tokenizer, **VibeToken**, that can efficiently encode any resolution into as few as 64 tokens, making our AR model 63× efficient than the baseline (LlamaGen) on a 1024×1024 resolution.

Abstract

*We introduce an efficient, resolution-agnostic autoregressive (AR) image synthesis approach that generalizes to arbitrary resolutions and aspect ratios, narrowing the gap to diffusion models at scale. At its core is **VibeToken**, a novel resolution-agnostic 1D Transformer-based image tokenizer that encodes images into a dynamic, user-controllable sequence of 32–256 tokens, achieving a state-of-the-art efficiency and performance trade-off. Building on **VibeToken**, we present **VibeToken-Gen**, a class-conditioned AR generator with out-of-the-box support for arbitrary resolutions while requiring significantly fewer compute resources. Notably, **VibeToken-Gen** synthesizes 1024×1024 images using only 64 tokens and achieves 3.94 gFID; by comparison, a diffusion-based state-of-the-art alternative requires 1,024 tokens and attains 5.87 gFID. In contrast to fixed-resolution AR models such as LlamaGen—whose inference FLOPs grow quadratically with resolution (≈11T FLOPs at 1024×1024)—**VibeToken-Gen** maintains a constant 179G FLOPs (63.4× efficient) independent of resolution. We hope **VibeToken** can help unlock the wide adoption of AR visual generative models in production use cases.*

1. Introduction

The landscape of visual generative modeling has been transformed by the advent of diffusion and autoregressive (AR) models [8, 12, 16, 18, 25, 31, 32, 42]. Diffusion models have emerged as the workhorse for production-scale image generation [9, 24, 28, 29]. In contrast, AR image synthesis [16, 32], despite achieving competitive performance, has seen limited deployment in production use cases. A key reason is flexibility: diffusion models generalize naturally to arbitrary resolutions and aspect ratios, while AR models struggle to do so. As a result, most existing AR literature targets fixed resolutions (e.g., 256×256, 512×512) and fails to adapt to arbitrary resolutions and aspect ratios [22, 33, 34, 39, 40]. A common workaround is to append super-resolution modules [42], but this introduces additional training complexity and computational cost (e.g., SDXL/Flux-upscaler) [9, 27].

We trace these scalability issues back to the image tokenizer [8, 21, 30, 32, 42]. As shown in Figure 2, conventional 2D CNN-based tokenizers (e.g., VQGAN [8]) produce a considerable number of tokens that grow linearly with image resolution, which in turn drives AR inference FLOPs to grow nearly quadratically due to self-attention [8, 21, 32]. The need for length generalization in next-token prediction compounds this burden. Recent 1D Transformer-based image tokenizers offer stronger compression and partially alleviate scaling challenges, but they still struggle to generalize across resolutions and aspect ratios, unlike their 2D counterparts [1, 4, 14, 17, 19, 23, 44].

This motivates us to ask the question: *"Can we encode images of arbitrary resolution into a fixed, small number of tokens and decode them back?"* If so, an AR generator could be trained at fixed dimensions, while the tokenizer shoulders most of the scaling burden.

In this work, we answer this by proposing novel *resolution-agnostic* image tokenization. We scale 1D image tokenizers to support arbitrary resolutions efficiently and introduce **VibeToken**. It encodes any input resolutions into a dynamic, user-controllable sequence of 32–256 tokens and decodes to any target resolution (input and output resolutions may differ). To achieve this, we systematically analyze and scale the architecture, yielding a truly flexible 1D tokenizer via: (1) dynamic patch embeddings that adapt to input resolutions, (2) variable image token representations, (3) efficient positional embeddings robust to resolution and aspect-ratio changes, and (4) pretraining strategies designed to promote out-of-distribution resolution generalization. This strategy confers both high compression and strong resolution generalization, and natively supports super-resolution.

Building on **VibeToken**, we present **VibeToken-Gen**, an efficient class-conditioned AR generator and inference pipeline that, for the first time, scales gracefully to arbitrary resolutions and aspect ratios without resolution-specific, compute-intensive training. We train **VibeToken-Gen** on ImageNet-1k with a training-time token budget comparable to (or smaller than) fixed-resolution AR baselines (e.g., LlamaGen at 256×256). We found that 64 tokens are sufficient for **VibeToken-Gen** to synthesize arbitrary-resolution images (including 1024×1024). Crucially, **VibeToken-Gen** requires a constant 179 GFLOPs at inference for any resolution, whereas LlamaGen scales to ≈11 TFLOPs at 1024×1024.

As a result, **VibeToken-Gen** offers substantial efficiency gains over existing AR models and points toward production-grade, resolution-generalist AR modeling. To summarize, our **key contributions are following:**

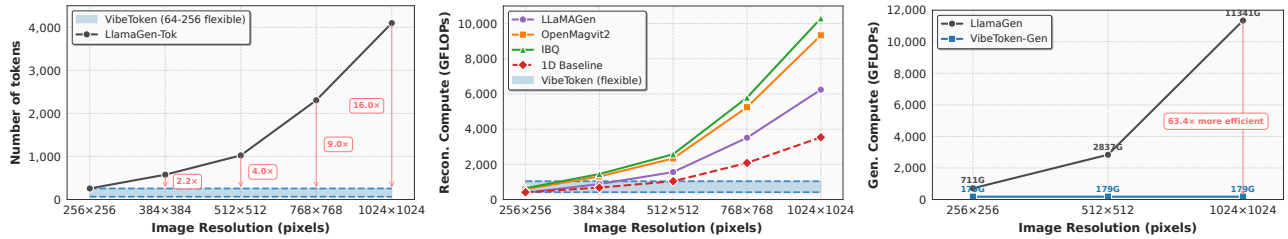
- We propose a resolution-agnostic image tokenizer that enables efficient, scalable AR image generation across arbitrary resolutions and aspect ratios.
- We scale 1D tokenizers to generalize to out-of-distribution resolutions and natively support image super-resolution.
- **VibeToken** achieves state-of-the-art reconstruction among 1D tokenizers while supporting dynamic-length (32–256) encoding.
- **VibeToken-Gen** maintains a constant 179 GFLOPs at inference, whereas prior AR models' FLOPs grow quadratically with resolution.
- **VibeToken-Gen** is faster (**5.14**×) and better (**33%**) than the diffusion counterpart (NiT) for 1024×1024 synthesis (0.21 s vs 1.08 s), both achieving gFID of 3.94 and 5.87, respectively [1].

2. Related Works

Autoregressive image generation. Visual AR models learn the likelihood of discrete visual tokens, typically in raster-scan order (i.e., row-by-row), through next-token prediction. Early approaches, such as VQGAN+Transformer [8], ViT-VQGAN [41], Parti [42], RQ-Transformer [15], and LlamaGen [16], encode images into a discrete latent space and perform AR modeling therein. Despite strong fidelity, scaling is hindered by token growth at higher resolutions and the quadratic attention cost with sequence length. Recent variants improve performance by altering factorization or ordering (e.g., scale-wise AR in VAR [34]; randomized orders in RAR [45]), achieving speed/quality gains. However, most AR systems are still trained at fixed, relatively low resolutions (256 × 256 or 512 × 512), and they struggle to generalize to arbitrary resolutions and aspect ratios due to rapidly increasing token counts and compute.

Image tokenizers. Tokenizers determine the sequence length, which in turn affects the feasibility of AR modeling. Classical designs quantize 2D VAE latents with vector codebooks, with many variants exploring quantization strategies such as lookup-free quantization (LFQ) [32] and multi-vector quantization (MVQ) [22]. Recently, DC-AR [40] at-

[1] **VibeToken** can be repurposed for diffusion or other form of AR modeling to achieve even better gains. Please see the future work and limitation section.



(a) **VibeToken** keeps token counts to 2-16× low as resolution increases.

(b) **VibeToken** keeps FLOPs significantly low and adjustable.

(c) **VibeToken-Gen** significantly lowers (63×) compute requirements.

Figure 2. **Compute comparisons.** Across resolution and metrics, **VibeToken** achieves strong efficiency (fewer tokens/FLOPs) compared to 2D baselines. This, in turn, leads to **VibeToken-Gen** being significantly more efficient with a constant 179 GFLOPs.

tempts to improve compression rates up to 32×, but it still requires 1024 tokens for 1024×1024 resolution images. 2D tokenizers often underperform in terms of compression at high resolutions. This has motivated Transformer-based 1D tokenizers that learn compact non-spatial latents with stronger compression ratios, including TiTok [44], TA-TiTok [14], One-D-Piece (1DP) [23], DetailFlow [19], and related methods [1, 4, 7]. These methods ease AR training by shortening sequences, but (unlike 2D grid tokenizers) they lack built-in spatial inductive bias and typically assume fixed training resolutions, which limits their generalization across high resolutions and aspect ratios.

Generalization to arbitrary resolutions. Resolution scalability is crucial for production use. Diffusion-based models have made notable progress: FiT-v1 [20]/FiT-v2 [35] improve cross-resolution generalization, and NiT [36] trains at native resolutions via architectural changes to DiT [26], scaling to arbitrary resolutions within an ImageNet-1k [6] compute budget. These advances, however, do not directly transfer to AR pipelines because AR compute scales with token length, which itself grows with image size under conventional tokenizers. Consequently, diffusion remains the default in production.

Our work aims to bridge this gap for AR by introducing a *resolution-agnostic* 1D tokenizer that maintains a short and controllable token budget across resolutions, allowing an AR generator to operate with a constant inference-time compute budget while supporting arbitrary aspect ratios.

3. Preliminaries

Autoregressive image generation. Let $v \in \mathbb{R}^{3\times H\times W}$ be an image and let a visual tokenizer $\mathcal{E}$ map it to a sequence of *discrete* tokens $x = (x_1, \ldots, x_T) \in [V]^T$, where V is the vocabulary size and T may depend on height (H) and width (W) through $\mathcal{E}$. An autoregressive (AR) model factorizes the likelihood via the chain rule:

$$p(x_{1:T}) = \prod_{t=1}^{T} p_\theta(x_t \mid x_{<t}). \tag{1}$$

In practice, sequence length for arbitrary resolution generalization is handled with an end-of-sequence (EOS) token:

$$p(x_{1:T}, \langle \text{eos} \rangle) = \prod_{t=1}^{T+1} p_\theta(x_t \mid x_{<t}), \quad \text{s.t. } x_{T+1} = \langle \text{eos} \rangle. \tag{2}$$

The training maximizes the log-likelihood (equivalently, minimizes next-token cross-entropy). Transformers are the standard choice for p_θ. For a sequence of length T, self-attention has $\mathcal{O}(T^2)$ time and memory per layer during training. At inference, AR decoding takes T steps; with KV-caching, the marginal cost of each new token is $\mathcal{O}(T)$, yielding $\mathcal{O}(T^2)$ total attention cost (without caching it would be $\mathcal{O}(T^3)$). Since many tokenizers increase T with resolution, the computational burden escalates quickly. For a typical f-stride 2D VQ-VAE, $T = (H/f) \cdot (W/f)$; with f=16, at 256×256 and 1024×1024 resolutions, we get T=256 and T=4096, respectively. This substantially increases the cost of AR model training and inference, and complicates length generalization.

1D image tokenization. Fix patch size k. Patchify $v \in \mathbb{R}^{3\times H\times W}$ into $N = \frac{HW}{k^2}$ patches, project each to $\mathbb{R}^d$, prepend L learned *latent* tokens, and encode:

$$x^{\text{enc}} = \mathcal{E}_\theta(x_0) \in \mathbb{R}^{(N+L)\times d}, \qquad h = x^{\text{enc}}_{N+1:N+L} \in \mathbb{R}^{L\times d}.$$

Let the codebook be $C = [c_1 \cdots c_m] \in \mathbb{R}^{m\times d}$. The quantizer Q maps each latent to its nearest vector:

$$z = Q(h) \in [m]^L,$$

We apply a reparameterization trick so that the gradients flow to h and C. Then we concatenate quantized latents with N learned masked output tokens $y_{\text{mask}} \in \mathbb{R}^{N\times d}$:

$$u = \mathcal{D}_\phi([\, C(z) \,\|\, y_{\text{mask}} \,]) \in \mathbb{R}^{(L+N)\times d}.$$

Latent positions $u_{1:L}$ (corresponding to encoded tokens) serve only as conditioning and are **ignored by the pixel head and loss**. This yields compact discrete latents for AR modeling, while the decoder predicts pixels exclusively at the masked positions.

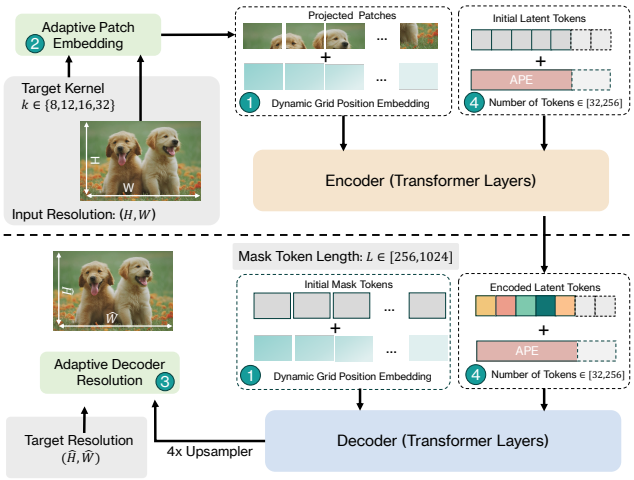


Figure 3. **Overview of VibeToken tokenizer.** VibeToken introduces four key components: 1) Dynamic grid position embedding, 2) Dynamic patch embedding, 3) Adaptive decoder resolution, and 4) Variable latent token-based resolution-agnostic encoding. These components provide full flexibility to 1D tokenizers, supporting arbitrary resolutions and compute controls.

Why this formulation struggles to scale? Grid-based 2D tokenizers make T grow linearly with increasing resolution. Although 1D Transformer tokenizers often achieve stronger compression, they inherit several bottlenecks akin to AR models:

- **Fixed resolution grid.** The triplet $(k, L, N_{\max})$ is typically fixed at pretraining. Out-of-distribution (OOD) resolutions require (i) ad-hoc positional-embedding interpolation (quality drop), or (ii) separate tokenizers for different resolutions/aspect ratios [44].
- **Quadratic attention cost;** Each self-attention layer forms a $s \times s$ similarity matrix ($s{=}N{+}L$), incurring $\mathcal{O}(s^2)$ FLOPs and memory per layer.
- **Rigid downstream generation.** Generators conditioned on discrete tokens inherit fixed or growing token budgets. Increasing resolution increases T, making generalization to arbitrary resolutions computationally expensive without explicit retraining.

These issues motivate a *resolution-agnostic*, compute-controllable tokenizer. In the next section, we introduce VibeToken, our first step toward a fully flexible 1D tokenizer that overcomes these obstacles.

4. Methodology

We first present VibeToken, a 1D tokenizer scaled to learn *resolution-agnostic* visual tokens that generalize to arbitrary resolutions and aspect ratios beyond the training distribution. We then introduce VibeToken-Gen, an AR generator (LlamaGen-style) adapted to VibeToken for efficient high-/arbitrary-resolution synthesis.

4.1. VibeToken

In Figure 3, we provide a high-level overview of our tokenizer. We focus on positional encoding, patch embedding, decoder adaptations for target resolution control, dynamic-length tokenization, and a training strategy that enforces resolution-agnostic behavior.

Dynamic grid positional embeddings. Varying resolutions imply a varying number of patches; absolute positional embeddings (APE) trained at a single grid often fail to extrapolate. Axial RoPE embeddings [11] help, but can be suboptimal or costly when made layer-wise learnable (see Table 1). We instead adopt a lightweight *dynamic grid position embedding*, inspired by NaViT [5] and ViTAR [10]. Let $\mathbf{G} \in \mathbb{R}^{d \times T_H^{\max} \times T_W^{\max}}$ be a learned grid of embeddings (we use $T_H^{\max}{=}T_W^{\max}{=}32$). For an input lattice with

$$T_H = \left\lceil \frac{H}{k_h} \right\rceil, \qquad T_W = \left\lceil \frac{W}{k_w} \right\rceil,$$

we obtain position codes by differentiable resizing operation (bilinear or bicubic) of $\mathbf{G}$ to (T_H, T_W) as:

$$\widehat{\mathbf{G}} \;=\; \mathrm{resize}\big(\mathbf{G};\, T_H,\, T_W\big) \;\in \mathbb{R}^{d \times T_H \times T_W},$$

then flatten to a sequence and add to patch embeddings. Latent (non-spatial) tokens utilize a separate regular APE, as we have a fixed number of latent tokens. Dynamic grid position embedding preserves inductive bias across resolutions/aspect ratios with negligible overhead and removes the fixed-length constraint of prior 1D tokenizers. As noted in Table 1, it yields a ∼33% FLOPs reduction without loss in quality compared to learnable axial RoPE.

Adaptive patch embedding. The length of the sequence s scales with the number of patches. Larger patches reduce computation, but risk detail loss; fixed-k training does not generalize effectively. Inspired by FlexiViT [2], we enable *variable* patch sizes $k \in [k_{\min}, k_{\max}]$ at *inference and training*.

Let $p_k \in \mathbb{R}^{3k^2}$ be a vectorized $k{\times}k$ RGB patch and let $W_{k_{\max}} \in \mathbb{R}^{d \times 3k_{\max}^2}$ be the *base* projection (*i.e.* kernel of the patch embedding layer). We define a *weight-resizing* operator as:

$$R_{k \leftarrow k_{\max}} \in \mathbb{R}^{3k^2 \times 3k_{\max}^2}$$

that maps the weights of the $k_{\max}$-grid to the k-grid (*e.g.* bilinear interpolation per channel applied $W_{k_{\max}}$). Hence, we obtain the differentiable resized kernel of patch embedding as:

$$W_k \;=\; W_{k_{\max}}\, R_{k \leftarrow k_{\max}}, \qquad e_{\text{patch}}(p_k) \;=\; W_k\, p_k + b.$$

Thus, only $W_{k_{\max}}, b$ are learned; all other W_k are derived on-the-fly, avoiding per-k reparameterization and yielding

Model	GFLOPs (1024^2)	rFID↓ 256^2	rFID↓ 1024^2
VAE-based small scale study			
V0 (TiTok-S-128 baseline)	–	1.862	–
w/ RoPE	299†	2.934	280.69
w/ Learnable RoPE	445†	1.544	93.49
w/ Dynamic Grid Embedding	299	1.707	131.20
+ Adaptive Patch Embedding	90	**1.366**	**5.38**
VibeToken-LL (Learnable RoPE)	1555†	0.46	2.57
VibeToken-LL (Grid)	1039	**0.40**	**2.40**

Table 1. **Ablation of positional embedding and adaptive patch embedding.** † indicates that FLOPs for RoPE-based models will be slightly higher due to issues in the custom functions calculations. Ablations are conducted in a VAE setting on a small encoder-decoder model for 200k iterations only. VibeToken-LL represents the final proposed tokenizer, we only perturb position embeddings.

consistent features across k. In practice, too large k can affect fine details if d is small; we cap $k_{\max}$=32 and *train* over $k \in \{8, 12, 16, 32\}$ to promote robustness.

Adaptive decoder resolution. Traditionally, the decoder predicts the fixed values of the k pixels per masked token. Instead, we decode at a *fixed* patch size $k_{\max} = 4k$ to an intermediate 4× high resolution and then introduce an extra downscaling 2D-CNN layer to get the desired target resolution. Notably, this downsampling layer can be adjusted to get the desired output resolution similar to adaptive patch embedding. Let the latent length be L and the spatial grid be (T_H, T_W). The decoder first predicts patch-wise pixels at resolution:

$$H' = T_H\, k_{\max}, \qquad W' = T_W\, k_{\max},$$

producing $\tilde{v} \in \mathbb{R}^{3\times H'\times W'}$. A lightweight learned downscaling 2D-Conv layer $\mathcal{D}_{H,W}$ then maps

$$\hat{v} \;=\; \mathcal{D}_{H,W}(\tilde{v}) \;\in\; \mathbb{R}^{3\times H\times W}. \tag{3}$$

Similar to adaptive patch embedding, we can control the kernel size of this layer to get the desired target resolution (H, W). This decouples decoding from the input patch size and enables native 4× super-resolution *without* a separate upscaler.

Dynamic-length tokenization. Images vary in complexity; a fixed latent length L is either wasteful or insufficient for ideal compression. Tail-token drop (e.g., One-D-Piece [23]) improves this tolerance and introduces dynamic length tokenization; however, it still trains the encoder at a fixed maximum length, potentially creating a quality gap during reconstructions. We instead *train* both encoder and decoder on uniform *sampled* lengths $L \sim \mathcal{P}(L)$ with $L \in [L_{\min}, L_{\max}]$ (e.g., 32–256). Given a target L, the encoder produces L latent tokens directly; the decoder consumes exactly L, with no padding. This yields strong quality–compression trade-offs (e.g., 1024×1024 → 64 tokens) and seamless compute control at inference, leading to state-of-the-art trade-off between number tokens *vs.* rFID (see Figure 4).

Training strategy. We train on images between 256×256 and 512×512 across aspect ratios {1:1, 1:2, 2:1, 2:3, 3:2}, sampling patch sizes $k \in \{8, 12, 16, 32\}$ such that the number of spatial tokens $N \leq 1024$. For each sample, we draw an *input* resolution $(H_{\text{in}}, W_{\text{in}})$ and an independent *target* resolution $(H_{\text{out}}, W_{\text{out}})$, training the model to reconstruct at $(H_{\text{out}}, W_{\text{out}})$ from $(H_{\text{in}}, W_{\text{in}})$. We also sample the latent length uniformly $L \in [32, 256]$. For quantization, we adopt a multi-codebook VQ (MVQ) variant compatible with our variable-length setting; implementation details and codebook updates are provided in the Appendix.[2]

Summary. The combination of Dynamic-APE, adaptive patch embedding, adaptive decoder resolution, and dynamic-length training yields a *resolution-agnostic* tokenizer with strong compression and low overhead.

4.2. VibeToken-Gen

VibeToken provides small *resolution-agnostic* discrete token sequences with controllable length L. To test its downstream image generation using VibeToken a natural choice would be to adapt a LlamaGen-style AR model (UniTok head with MVQ compatibility) to operate on these discrete tokens while keeping training compute comparable to a fixed 256×256 resolution baseline.

Conditioning on target resolution/aspect ratio. Although VibeToken can decode to any (H, W), it can exhibit stretching artifacts, such as resizing a square image into a vertical/horizontal shape. We therefore condition the AR model on the desired (H, W) alongside the class label y, and the rest of the AR stack is unchanged:

$$c = \big[\,\mathrm{emb}(y) + \mathrm{MLP}((H, W)/\beta)\,\big],$$

where $emb : \mathcal{R}^1 \to \mathcal{R}^d$ and $MLP : \mathcal{R}^2 \to \mathcal{R}^d$ are projection layers, while $\beta = 1536$ is normalization constant.

Training and inference. We train on the same resolution/aspect ratio mix as VibeToken, sampling latent lengths $L \in \{64, 128, 256\}$ with class conditioning. Because L is small (e.g., ≤ 256 for up to 1024^2) and independent of

[2]Briefly, MVQ improves high-compression performance by distributing capacity across multiple codebooks.

Models	Dynamic		Tokens	rFID Performance @ Image Resolutions			
	Resolution	Tokens		256^2	512^2	1024^2	Arbitrary
2D Image Tokenizers							
LLaMAGen-Tok [32]	✓	✗	16x‡	2.19	0.70	2.01	2.48
Open-MAGVIT-v2 [21]	✓	✗	16x‡	1.17	0.50	1.32	1.52
IBQ [30]	✓	✗	16x‡	**0.97**	**0.40**	**1.26**	**1.42**
DA-HT [40]	✓	✗	32x‡	1.60	0.83	N/A	N/A
1D Image Tokenizers							
TiTok-B [44]	✗	✗	64/128	1.70	1.52†	–	–
VAR [34]	✗	✗	680	0.90	–	–	–
ImageFolder [17]	✗	✗	286	0.80	–	–	–
DetailFlow [19]	✗	✗	512	0.55	–	–	–
UniTok [22]	✗	✗	256	**0.33**	–	–	–
Instella [37]	✓	✗	128	N/A	1.32	N/A	–
FlexTok [1]	✗	✓	1–256	1.08	–	–	–
One-D-Piece-L [23]	✗	✓	1–256	1.08	–	–	–
VibeToken–SL	✓	✓	32–256	0.43	0.55	2.45	4.21
VibeToken–LL	✓	✓	32–256	0.40	**0.51**	**2.40**	**3.60**

Table 2. **Tokenizer comparison across resolutions.** ✓ and ✗ indicates whether specific capability is supported or not, N/A indicates either model not available or failed to reproduce, – denotes method does not work, and † indicates resolution-specific (independently) trained model. Importantly, ‡ represents the compression ratios (f). Hence, token count ($(H \cdot W)/f^2$) varies *w.r.t.* target resolutions.

Model	Type	gFID at arbitrary low resolutions								
		1:1 256×256	2:3 256×368	3:2 368×256	3:4 368×512	4:3 512×368	1:2 256×512	2:1 512×256	1:1 512×512	Average
EDM2-L [13]	Diff.	70.19	36.29	32.70	3.29	3.19	18.11	14.30	1.92	22.50
FiTv2-XL [35]		**2.26**	6.28	6.30	155.97	176.66	27.31	35.36	259.11	83.66
NiT-XL [36]		2.27	**4.28**	**4.00**	**2.81**	**3.00**	**9.60**	**6.04**	**1.80**	**4.22**
VibeToken-Gen (B)	AR	7.62	9.19	7.87	7.63	7.46	15.83	10.08	7.45	9.14
VibeToken-Gen (XXL)		3.62	5.60	4.22	4.00	3.78	12.47	7.76	3.60	5.63

Model	Type	gFID at higher resolutions								
		1:1 512×512	2:3 512×768	3:2 768×512	3:4 768×1024	4:3 1024×768	1:2 512×1024	2:1 1024×512	1:1 1024×1024	Average
EDM2-L [13]	Diff.	1.92	5.62	6.04	31.03	39.54	17.54	19.87	64.32	23.23
FiTv2-XL (highres)† [35]		2.93	20.61	29.49	155.97*	176.66*	163.91	185.89	259.11*	124.32
NiT-XL [36]		**1.80**	**4.45**	4.47	4.89	5.14	12.23	9.57	5.87	6.05
VibeToken-Gen (B)‡	AR	7.62	8.97	7.64	7.57	7.46	15.23	10.07	7.36	8.99
VibeToken-Gen (XXL)‡		3.69	5.32	**4.01**	**4.05**	**3.84**	**11.91**	**7.88**	**3.54**	**5.53**

Table 3. **gFID across aspect ratios and resolutions (single view).** Top: low resolutions (256–512); bottom: high (512–1024). The first header line shows the aspect ratio, and the second line lists the corresponding pixel resolution. ∗ results are borrowed from low-resolution experiments due to the days of inference computing requirements. † FiT-v2 has high resolution trained separately for 512×512 and plus. ‡ VibeToken-Gen utilizes the native super resolution of our tokenizer.

(H, W), VibeToken-Gen maintains a *constant* inference-time FLOPs budget across resolutions; compute is governed primarily by L and the AR depth, **not by target resolution**.

5. Experimental Results

We detail implementation and evaluation results for our tokenizer and the class-conditioned autoregressive generator. We report reconstruction and generation performance across resolutions and analyze efficiency.

5.1. Image Reconstruction

Experimental setup. We train two variants of VibeToken: **VibeToken-SL** (small encoder, large decoder) and **VibeToken-LL** (large encoder, large decoder). Both are trained from scratch on ImageNet1k with mixed resolutions between 256×256 and 512×512; 60% of the samples are square (256^2 or 512^2) and the remainder are drawn from non-square aspect ratios. We use a batch size of 64 for 600k iterations with cosine decay and peak learning rate 1e−4.

Tokens	gFID	
	w/o cfg	w/ cfg
256	39.32	9.81
128	37.68	9.02
64	**33.15**	**8.42**

(a) Impact of token length.

Tokens	Generalist	gFID	
		w/o cfg	w/ cfg
64	✗	33.15	**8.42**
	✓	**31.14**	9.37
128	✗	**37.68**	**9.02**
	✓	39.50	10.58

(b) Impact of resolution-agnostic training.

Model	Tokens	Generalist	gFID	
			w/o cfg	w/ cfg
LlamaGen	576	✗	33.44	**7.15**
VibeToken-Gen	**64**	✓	**31.14**	9.37

(c) Comparison with LlamaGen (fixed resolution).

Table 4. **Ablation study on ImageNet 256×256 generations.** We perform a study on (a) the number of tokens, (b) generalist training, and (c) compare with baseline LlamaGen. All models are GPT-B variants and were trained for 100 epochs with a batch size of 256.

Unless noted otherwise, we train for dynamic latent lengths $L \in [32, 256]$. We adopt MVQ with 8 codebooks of size 4096 (effective vocabulary 32,768). All models are trained on a single node with 8×H100 GPUs; further hyperparameters are in the Appendix.

Evaluation protocol. We assess reconstruction on three different *i.i.d.* and *o.o.d.* regimes. We report rFID (lower is better) and vary L where applicable.

1. ImageNet-1k (i.i.d.): 50k validation images at 256^2 and 512^2.
2. FFHQ (high-res): 10k images at 1024^2.
3. Arbitrary aspect ratios (OOD): stress tests at aspect ratios $\{1:2, 2:3, 3:4, 9:16, 16:9, 4:3, 3:2, 2:1\}$ with resolutions between 512^2 and 1024^2 using FFHQ.

Main results. Table 2 compares **VibeToken** with 2D and 1D tokenizers across resolutions. Unlike prior 1D tokenizers (which generally assume fixed training grids), **VibeToken** *natively* supports arbitrary resolutions and aspect ratios while retaining short, dynamic token sequences. Concretely, **VibeToken-LL** attains **0.40** rFID at 256^2, **0.51** at 512^2, **2.40** at 1024^2, and **3.60** on arbitrary-resolution stress tests; **VibeToken-SL** achieves 0.43/0.55/2.45/4.21 on the same settings. Among 1D baselines, UniTok reports a strong 0.33 rFID at 256^2 but does not support dynamic resolution; several other 1D methods lack results beyond 256^2. Against 2D tokenizers, **VibeToken** is competitive on 256^2–512^2 (e.g., IBQ: 0.97/0.40) while trading a modest performance drop at 1024^2, **VibeToken** dramatically reduces token requirements at arbitrary high resolutions. Figure 4 shows the dynamic compression *vs.* rFID performance trade-off. With as few as **64 tokens**, **VibeToken** matches or surpasses prior dynamic-length 1D tokenizers at 256^2; increasing to 128–256 tokens yields further gains while preserving the ability to decode at arbitrary target resolutions. We provide detailed ablations and native super-resolution performance in the appendix.

Efficiency analysis. Figure 2 plots tokenizer inference FLOPs across resolutions. 2D tokenizers scale roughly linearly with pixel count: e.g., IBQ grows from ~ 0.64 T to ~ 10.30 T FLOPs from 256^2 to 1024^2 resolution. In contrast, **VibeToken maintains constant ~ 1.04 T FLOPs** (maximum), enabling a fixed compute budget independent of resolution. This stability, combined with dynamic lengths (L=64–256), yields state-of-the-art throughput among flexible tokenizers. Moreover, while 2D tokenizers have a fixed 16×/ 32× compression, **VibeToken** achieves much higher effective compression as resolution increases: at 1024^2, images can be encoded with **64 – 256 tokens**, improving token efficiency by 16× **to** 64× relative to 2D counterparts; resulting into effective compression ratio of 64× to 128×.

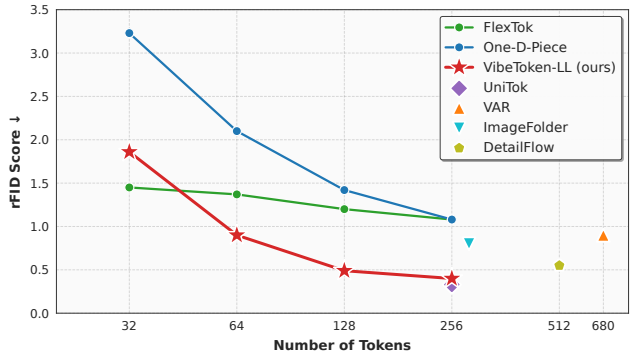


Figure 4. **Dynamic token length.** rFID vs. latent length L at 256^2. **VibeToken** is the only model that pairs competitive quality with *resolution generalization*.

5.2. Image Generation

We evaluate the *resolution-agnostic* tokens produced by **VibeToken** by training class-conditioned AR generators that can support arbitrary target resolutions. Because scaling prior AR models across resolutions with existing tokenizers is compute-intensive and impractical in academic settings, we compare against (i) open-source multi-resolution diffusion models and (ii) fixed-resolution AR baselines.

Experimental setup. We train **VibeToken-Gen** on mixed resolutions from 256^2 to 512^2 within an ImageNet-1k compute envelope. We use Query–Key LayerNorm for stability, select **VibeToken-LL** as the default tokenizer, and keep the rest of the LlamaGen-style stack unchanged. Because

Model	Type	Params	Arbitrary Resolutions	256×256			512×512		
				Tokens	FID	IS	Tokens	FID	IS
MaskGIT [3]	Mask.	177M	✗	256	4.02	355.6	1024	4.46	342
TiTok-S/B-128 [44]		287M	✗	128	2.48	–	128	2.13	–
MAGVIT-v2 [43]		307M	✗	256	1.78	319.4	1024	1.91	324.3
MaskBit [38]		305B	✗	256	1.52	328.6	–	–	–
VAR-d30 [34]	VAR	2B	✗	680	1.92	323.1	2240	2.63	303.2
VAR-d30-re [34]		2B	✗	680	1.73	350.2	–	–	–
GPT2-re [8]	AR	1.4B	✗	256	5.20	280.3	–	–	–
VIM-L-re [41]		1.7B	✗	1024	3.04	227.4	–	–	–
Open-MAGVIT2-XL [21]		1.5B	✗	256	2.33	271.77	–	–	–
LlamaGen-XXL [32]		1.4B	✗	576	2.34	253.91	–	–	–
UniTok-XXL [22]		1.4B	✗	256	2.51	216.7	–	–	–
RAR-XXL [45]		1.4B	✗	256	1.48	326.0	1024	1.66	295.7
Single AR model for all Resolutions									
VibeToken-Gen-B	AR	87M	✓	**64**	7.62	185.92	**64**	7.48	189.43
VibeToken-Gen-XXL		1.5B	✓	**64**	3.62	226.93	**64**	3.60	230.33

Table 5. **Comparison across models and resolutions.** Columns group 256×256 and 512×512 results (Tokens/FID/IS). Among many prior works on discrete image modeling, only **VibeToken-Gen** scales to arbitrary resolutions effortlessly.

VibeToken supplies small, resolution-agnostic sequences, the training compute remains comparable to (or lower than) a fixed 256×256 LlamaGen baseline. Additional details are in the Appendix.

Ablation study. Table 4 reports GPT-B ablations over 100 epochs. We find that **64 tokens** are enough and yield the best gFID. Notably, increasing the token length to 128/256 does not offer any gains. This highlights that not all tokens important for improving rFID are needed to improve gFID. Training a *generalist* (mixed-resolution) model introduces a small degradation relative to single-resolution training, but the **64-token** setting remains competitive. Guided by these findings, we train **VibeToken-Gen**-XXL generalist models with only 64 tokens and report the main results.

Main results. We benchmark class-conditional generation against both AR and diffusion families. Since prior AR models do not natively generalize across resolutions, our primary comparisons are to multi-resolution diffusion baselines. As shown in Table 3, **VibeToken-Gen** *smoothly* scales to higher and arbitrary resolutions and achieves near-SOTA gFID despite being trained within ImageNet-1k budgets. Concretely, it outperforms EDM2 and FIT-v2 across many resolution settings and is competitive with NIT. This demonstrates that AR models *can* be made resolution-generalist on academic compute by delegating scale handling to a resolution-agnostic tokenizer. Finally, when evaluated strictly at 256^2 or 512^2 against models trained only at those resolutions (Table 5), **VibeToken-Gen** can trail specialist baselines; we attribute this to the shared budget across resolutions during generalist training, which makes direct comparison to single-resolution specialists less informative. We provide qualitative examples in the appendix.

Efficiency analysis. AR models are notoriously expensive to scale, especially when token counts grow with resolution. **VibeToken-Gen** breaks this pattern by decoupling compute from (H, W): with a fixed latent length L, inference cost is governed by L and model depth—not by image resolution. As shown in Figure 2, **VibeToken-Gen**-XXL requires **179 G** *per forward step* with $L{=}64$ for *any* resolution,[3] whereas a fixed-resolution LlamaGen scales up to $\sim$11 T FLOPs to reach 1024^2 even in idealized settings. In wall-clock terms, a diffusion SOTA baseline (NiT) takes **1.08** s to synthesize a single 1024^2 image at **5.87** gFID, while **VibeToken-Gen-XXL** achieves **3.54** gFID with **0.22** s latency. These results underscore that resolution-agnostic tokens enable high-resolution AR generation with substantially lower and more predictable compute.

6. Conclusion

We present **VibeToken** and **VibeToken-Gen**, a practical recipe for scaling autoregressive (AR) image generation to *arbitrary* resolutions and aspect ratios under a fixed, user-controllable compute budget. The core idea is a *resolution-agnostic* 1D tokenizer (**VibeToken**) that produces short, variable-length sequences (64-256 tokens) independent of the resolutions. This is achieved through dynamic grid positional embeddings, adaptive decoding, and length-aware training. To our knowledge, **VibeToken** is the first 1D Transformer-based tokenizer to natively support arbitrary resolutions while retaining strong compression and reconstruction quality. Building on these tokens, **VibeToken-Gen** achieves competitive class-conditional generation across resolutions, with inference FLOPs governed by token length rather than pixel count, thereby narrowing the gap between AR and diffusion pipelines within academic compute budget.

[3]Numbers reported per forward pass without KV caching; The total generation cost is therefore higher and scales with the sequence length L.

References

[1] Roman Bachmann, Jesse Allardice, David Mizrahi, Enrico Fini, Oğuzhan Fatih Kar, Elmira Amirloo, Alaaeldin El-Nouby, Amir Zamir, and Afshin Dehghan. Flextok: Resampling images into 1d token sequences of flexible length. In *Forty-second International Conference on Machine Learning*, 2025. 2, 3, 6

[2] Lucas Beyer, Pavel Izmailov, Alexander Kolesnikov, Mathilde Caron, Simon Kornblith, Xiaohua Zhai, Matthias Minderer, Michael Tschannen, Ibrahim Alabdulmohsin, and Filip Pavetic. Flexivit: One model for all patch sizes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14496–14506, 2023. 4

[3] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11315–11325, 2022. 8

[4] Hao Chen, Ze Wang, Xiang Li, Ximeng Sun, Fangyi Chen, Jiang Liu, Jindong Wang, Bhiksha Raj, Zicheng Liu, and Emad Barsoum. Softvq-vae: Efficient 1-dimensional continuous tokenizer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28358–28370, 2025. 2, 3

[5] Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim M Alabdulmohsin, et al. Patch n'pack: Navit, a vision transformer for any aspect ratio and resolution. *Advances in Neural Information Processing Systems*, 36:2252–2274, 2023. 4

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3

[7] Shivam Duggal, Phillip Isola, Antonio Torralba, and William T Freeman. Adaptive length image tokenization via recurrent allocation. In *First Workshop on Scalable Optimization for Efficient and Adaptive Foundation Models*, 2024. 3

[8] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 2, 8

[9] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024. 2

[10] Qihang Fan, Quanzeng You, Xiaotian Han, Yongfei Liu, Yunzhe Tao, Huaibo Huang, Ran He, and Hongxia Yang. Vitar: Vision transformer with any resolution. *arXiv preprint arXiv:2403.18361*, 2024. 4

[11] 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

[12] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2

[13] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24174–24184, 2024. 6

[14] Dongwon Kim, Ju He, Qihang Yu, Chenglin Yang, Xiaohui Shen, Suha Kwak, and Liang-Chieh Chen. Democratizing text-to-image masked generative models with compact text-aware one-dimensional tokens. *arXiv preprint arXiv:2501.07730*, 2025. 2, 3

[15] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11523–11532, 2022. 2

[16] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37:56424–56445, 2024. 2

[17] Xiang Li, Kai Qiu, Hao Chen, Jason Kuen, Jiuxiang Gu, Bhiksha Raj, and Zhe Lin. Imagefolder: Autoregressive image generation with folded tokens. *arXiv preprint arXiv:2410.01756*, 2024. 2, 6

[18] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[19] Yiheng Liu, Liao Qu, Huichao Zhang, Xu Wang, Yi Jiang, Yiming Gao, Hu Ye, Xian Li, Shuai Wang, Daniel K Du, et al. Detailflow: 1d coarse-to-fine autoregressive image generation via next-detail prediction. *arXiv preprint arXiv:2505.21473*, 2025. 2, 3, 6

[20] Zeyu Lu, Zidong Wang, Di Huang, Chengyue Wu, Xihui Liu, Wanli Ouyang, and Lei Bai. Fit: Flexible vision transformer for diffusion model. *arXiv preprint arXiv:2402.12376*, 2024. 3

[21] Zhuoyan Luo, Fengyuan Shi, Yixiao Ge, Yujiu Yang, Limin Wang, and Ying Shan. Open-magvit2: An open-source project toward democratizing auto-regressive visual generation. *arXiv preprint arXiv:2409.04410*, 2024. 2, 6, 8

[22] Chuofan Ma, Yi Jiang, Junfeng Wu, Jihan Yang, Xin Yu, Zehuan Yuan, Bingyue Peng, and Xiaojuan Qi. Unitok: A unified tokenizer for visual generation and understanding. *arXiv preprint arXiv:2502.20321*, 2025. 2, 6, 8

[23] Keita Miwa, Kento Sasaki, Hidehisa Arai, Tsubasa Takahashi, and Yu Yamaguchi. One-d-piece: Image tokenizer meets quality-controllable compression. *arXiv preprint arXiv:2501.10064*, 2025. 2, 3, 5, 6

[24] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 2

[25] Maitreya Patel, Changhoon Kim, Sheng Cheng, Chitta Baral, and Yezhou Yang. Eclipse: A resource-efficient text-to-image prior for image generations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9069–9078, 2024. 2

[26] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 3

[27] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 2

[28] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024. 2

[29] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022. 2

[30] Fengyuan Shi, Zhuoyan Luo, Yixiao Ge, Yujiu Yang, Ying Shan, and Limin Wang. Scalable image tokenization with index backpropagation quantization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16037–16046, 2025. 2, 6

[31] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 2

[32] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024. 2, 6, 8

[33] NextStep Team, Chunrui Han, Guopeng Li, Jingwei Wu, Quan Sun, Yan Cai, Yuang Peng, Zheng Ge, Deyu Zhou, Haomiao Tang, et al. Nextstep-1: Toward autoregressive image generation with continuous tokens at scale. *arXiv preprint arXiv:2508.10711*, 2025. 2

[34] 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. 2, 6, 8

[35] ZiDong Wang, Zeyu Lu, Di Huang, Cai Zhou, Wanli Ouyang, et al. Fitv2: Scalable and improved flexible vision transformer for diffusion model. *arXiv preprint arXiv:2410.13925*, 2024. 3, 6

[36] Zidong Wang, Lei Bai, Xiangyu Yue, Wanli Ouyang, and Yiyuan Zhang. Native-resolution image synthesis. *arXiv preprint arXiv:2506.03131*, 2025. 3, 6

[37] Ze Wang, Hao Chen, Benran Hu, Jiang Liu, Ximeng Sun, Jialian Wu, Yusheng Su, Xiaodong Yu, Emad Barsoum, and Zicheng Liu. Instella-t2i: Pushing the limits of 1d discrete latent space image generation. *arXiv preprint arXiv:2506.21022*, 2025. 6

[38] Mark Weber, Lijun Yu, Qihang Yu, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. Maskbit: Embedding-free image generation via bit tokens. *arXiv preprint arXiv:2409.16211*, 2024. 8

[39] Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, et al. Vila-u: a unified foundation model integrating visual understanding and generation. *arXiv preprint arXiv:2409.04429*, 2024. 2

[40] Yecheng Wu, Han Cai, Junyu Chen, Zhuoyang Zhang, Enze Xie, Jincheng Yu, Junsong Chen, Jinyi Hu, Yao Lu, and Song Han. Dc-ar: Efficient masked autoregressive image generation with deep compression hybrid tokenizer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18034–18045, 2025. 2, 6

[41] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021. 2, 8

[42] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022. 2

[43] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023. 8

[44] Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. *Advances in Neural Information Processing Systems*, 37:128940–128966, 2024. 2, 3, 4, 6, 8

[45] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Randomized autoregressive visual generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18431–18441, 2025. 2, 8