CHANNELTOK: EFFICIENT FLEXIBLE-LENGTH VISION TOKENIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Leading flexible vision tokenizers achieve SOTA quality at an extreme cost, relying on parameter-heavy backbones and slow, multi-step generative decoders. We depart from this complex, spatial-token paradigm and introduce a simple, lightweight, and fast channel-wise flexible-length tokenizer. Our method treats each latent channel as a visual token, enabling a parameter-efficient CNN-Transformer hybrid backbone. Furthermore, employing a stochastic tail-dropping paradigm during training naturally forces channels to organize by semantic importance. This allows for flexible compression at inference by simply retaining the first k channels, and naturally enables variable-length autoregressive image generation. We validate our approach through extensive experiments on ImageNet, demonstrating consistent quality across diverse token budgets. The results establish a new quality-efficiency frontier: our model achieves state-of-the-art perceptual quality (rFID 2.92) while being 8.6× faster in decoding and 2.1× smaller (159M params) than the next-best alternative. Our work establishes channel-wise tokenization as a powerful and practical paradigm for efficient visual representation.

1 INTRODUCTION

Vision tokenization serves as a foundational component in discrete generative image models, compressing images from pixel space into compact, semantically rich representations in latent space. While early generative representations for images focused on continuous-valued autoencoders (Hinton & Salakhutdinov, 2006) and VAEs (Kingma & Welling, 2013), the pursuit of more efficient and discrete representations led to vector quantization, first via explicit codebooks in VQ-VAE (van den Oord et al., 2017) and VQ-GAN (Esser et al., 2021), and more recently through lookup-free methods such as FSQ (Mentzer et al., 2023), LFQ (Yu et al., 2023), and BSQ (Zhao et al., 2024).

Efficiency in vision tokenization spans multiple dimensions like compression ratio, reconstruction fidelity, semantic richness, encoding-decoding latency, and downstream task performance. Until recently, however, another critical dimension remained largely unexplored: *flexible-length tokenization* that adjusts representation length based on visual complexity. This need has become increasingly urgent in the era of large-scale vision models, where compute budgets constrain deployment and the inherent variability in visual data suggests that not all images require equal representational capacity.

In this work, we introduce a simple *channel-wise* approach for training a flexible-length tokenizer. Our approach is based off a lightweight VQGAN (Esser et al., 2021) based autoencoder where each channel slice forms a visual token. However, unlike standard architectures, our latent features are ordered from low- to high-level. On each step, the encoder maps an image onto the latent space, a random number k of features is chosen, and the decoder must reconstruct the image using only the k lowest-level features. From this process, a coarse-to-fine hierarchy of features naturally emerges in which "low-level" features represent concepts and shapes, and "high-level" features represent details.

By leveraging the inherent structure along the channel dimension, our architecture avoids the complexity of parameter-heavy, multi-step generative decoders (Bachmann et al., 2025; Mao et al., 2025). Fig. 1 shows our method strikes a favorable balance across the quality-efficiency frontier: competitive reconstruction quality (rFID 2.92) while being the smallest (159M parameters) and among the fastest high-fidelity flexible tokenizers. This balance of efficiency and fidelity makes our method ideally suited for deployment scenarios where size, speed, and quality collectively matter, such as edge devices and real-time applications.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

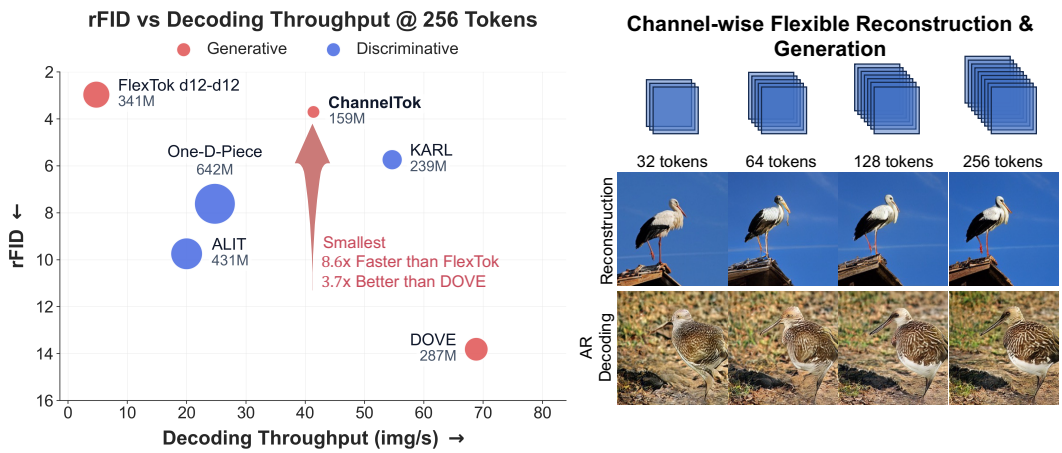

Figure 1: **Quality-efficiency comparison.** Reconstruction fidelity (rFID), decoding throughput, and model size across recent tokenizers. Our method achieves state-of-the-art rFID while being the smallest and among the fastest decoders. **Right:** Channel-wise reconstruction and AR generation.

Our contributions include: (1) a channel-wise flexible tokenization mechanism that dynamically allocates latent channel capacity based on image complexity; (2) a lightweight VQGAN-based architecture optimized for parameter efficiency and high-throughput encoding–decoding; (3) training strategies and regularization for stable learning in the channel-dimension tokenization regime; and (4) a demonstration that channel-wise token ordering transfers to autoregressive generation, where a LlamaGen model trained on our tokens produces coherent images at variable token budgets, suggesting channel-wise tokenization as a promising direction for efficient AR generation.

2 Background & Related Work

Recent works address flexible tokenization through various mechanisms. CAT (Shen et al., 2025) employs an external LLM-based complexity evaluator to select among compression ratios, realised by a nested VAE with variable-resolution latents. Other approaches enforce nested token hierarchies through tail-token dropping (Bachmann et al., 2025; Miwa et al., 2025) and block-wise causal masking (Yan et al., 2024). These Matryoshka-inspired (Kusupati et al., 2022) methods demonstrate a semantic hierarchy from coarse to fine representations through explicit token ordering. ALIT (Duggal et al., 2024) instead uses recurrence to iteratively adjust token lengths over multiple roll-outs, rather than nesting multiple granularities within a single pass. As a further departure, KARL (Duggal et al., 2025) predicts token count in one forward pass using Kolmogorov Complexity as a proxy for minimal description length. However, it uses upside-down reinforcement learning training and is sensitive to the reconstruction threshold controlling its token–quality trade-off.

While these methods expand the notion of "flexibility" to include reconstruction fidelity, downstream performance, and latent compressibility, industry-scale deployment demands additional considerations: parameter efficiency and inference speed. Many approaches rely on sophisticated, parameter-heavy architectures with large ViT backbones (Tian et al., 2024; Mao et al., 2025; Bachmann et al., 2025). Despite compelling reconstruction fidelity, some methods require multi-step generative decoders (Bachmann et al., 2025; Mao et al., 2025), adding inference overhead. Others limit compression to fixed token buckets (Duggal et al., 2024) or train on hundreds of millions of samples (Yan et al., 2024). These approaches require extensive training recipes, including multi-stage training pipelines or prolonged training schedules, to maintain quality in low-token regimes. Inspired by recent findings that latent channels naturally exhibit a coarse-to-fine hierarchy (Zhuang et al., 2025; Yao & Wang, 2025), we address these limitations by revisiting flexible tokenization along the channel dimension rather than the spatial dimension. This shift enables adaptive token granularity with a substantially simpler and more parameter-efficient design, without relying on explicit token hierarchies or complex architectural mechanisms.

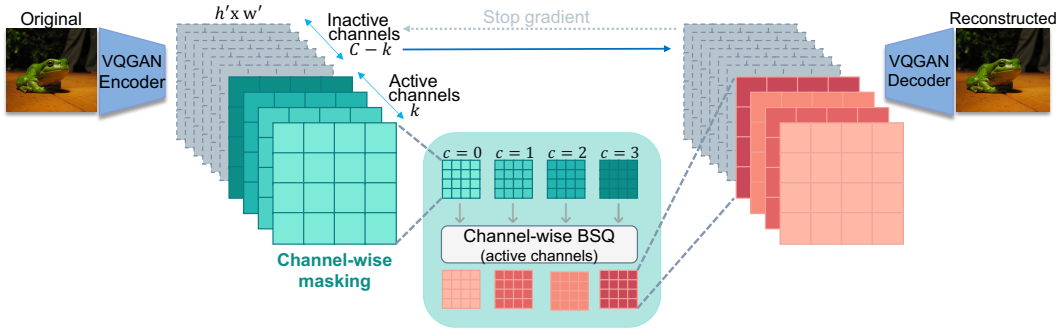

Figure 2: **Overview of our channel-wise flexible tokenizer.** The encoder compresses the image into a latent representation $\mathbf{z} \in \mathbb{R}^{C\times h\times w}$. During training, we adaptively mask channels by retaining only the first k active channels (shown in teal) while stopping gradients through inactive channels (shown in gray). Each active channel is independently quantized using Binary Spherical Quantization (BSQ). The decoder reconstructs the image from the quantized active channels and the masked inactive channels. At inference, varying k enables flexible compression rates without retraining.

3 Method

We introduce a channel-wise adaptive tokenizer that dynamically selects the number of latent channels to retain for each input. Our approach employs a VQGAN-inspired (Esser et al., 2021) autoencoder where each latent channel functions as a discrete visual token. An overview appears in Fig. 2.

3.1 Preliminaries

Let $\mathbf{x} \in \mathbb{R}^{H\times W\times 3}$ denote an input image. Our tokenizer comprises an encoder E_θ, quantizer Q_ϕ, and decoder D_ψ. The encoder compresses $\mathbf{x}$ into latent representation $\mathbf{z} = E_\theta(\mathbf{x}) \in \mathbb{R}^{C\times h\times w}$, where C is the channel dimension and h, w are spatial dimensions. Unlike spatial tokenization methods treating each spatial location as a token, we operate along the channel dimension where each channel slice $\mathbf{z}_c \in \mathbb{R}^{h\times w}$ constitutes a visual token. This design enables a natural coarse-to-fine hierarchy without explicit ordering constraints.

3.2 Encoder-Decoder Architecture

Encoder. Building on VQGAN (Esser et al., 2021), our encoder applies convolutional blocks with residual connections and downsampling layers. Multi-head self-attention blocks at resolutions $\{32, 16, 8\}$ capture global dependencies. The encoder reduces spatial resolution by factor 16 (from 256×256 to 16×16) while expanding to $C = 512$ channels, yielding $\mathbf{z} \in \mathbb{R}^{512\times 16\times 16}$.

Channel as token. Each of the C channels encodes distinct visual information at varying abstraction levels. We treat the channel dimension as the token dimension, where $\mathbf{z}_c$ for $c \in \{1, \ldots, C\}$ represents the c-th visual token. This formulation differs fundamentally from spatial tokenization: rather than encoding local patches, each channel captures global patterns across the entire spatial extent, with earlier channels encoding coarse structure and later channels refining details.

Decoder. The decoder mirrors the encoder with upsampling blocks and multi-head self-attention at resolutions $\{8, 16, 32\}$. It reconstructs the image $\hat{\mathbf{x}} = D_\psi(\mathbf{q})$ from quantized representation $\mathbf{q}$ with inactive channels set to zero.

3.3 Adaptive Channel Masking

To enable flexible tokenization, we introduce a stochastic prefix channel masking mechanism during training. This encourages the model to learn an ordered representation where information importance decreases along the channel dimension.

Mask generation and application. For each training sample, we uniformly sample mask retention ratio $t \sim \mathcal{U}(t_{\min}, t_{\max})$ with $t_{\min} = 0.002$ and $t_{\max} = 1.0$. The number of active channels is

Algorithm 1 Adaptive Channel Masking.

Require: Latent $\mathbf{z} \in \mathbb{R}^{C\times h\times w}$, mask probability p_{mask}, range $[t_{\min}, t_{\max}]$
Ensure: Masked latent $\mathbf{z}_{\text{input}}$, mask $\mathbf{M}$, active channels k
Sample apply_mask $\sim$ Bernoulli(p_{mask})
if apply_mask $= 1$ **then**
  Sample $t \sim \mathcal{U}(t_{\min}, t_{\max})$; $k \leftarrow \max(1, \min(\lfloor t \cdot C \rfloor, C))$
  $\mathbf{M} \leftarrow [\mathbf{1}_{h\times w}$ if $c \le k$ else $\mathbf{0}_{h\times w}]_{c=1}^{C}$ *// Prefix mask: channels $1{:}k$ active, $k{+}1{:}C$ zeroed*
  $\mathbf{z}_{\text{input}} \leftarrow \mathbf{M} \odot \mathbf{z} + (1-\mathbf{M}) \odot \text{sg}(\mathbf{z})$ *// All channels reach the decoder; only the first k get gradients*
else
  $\mathbf{M} \leftarrow \mathbf{1}_{C\times h\times w}$; $k \leftarrow C$; $\mathbf{z}_{\text{input}} \leftarrow \mathbf{z}$ *// No masking: full-context reconstruction over all C channels*
end if
return $\mathbf{z}_{\text{input}}, \mathbf{M}, k$

$k = \lfloor t \cdot C \rfloor$, clamped to $[1, C]$. We construct binary mask $\mathbf{M} \in \{0,1\}^{C\times h\times w}$ as:

$$\mathbf{M}_c = \begin{cases} 1 & \text{if } c \le k \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The mask will be used to stochastically drop the tail of the feature tensor, promoting hierarchical organization where critical information concentrates in early channels, and fine details in later channels.

During training, we randomly choose between applying masking or using the full latent $\mathbf{z}$, each with equal probability. For masked channels ($c > k$), we apply stop-gradient to prevent encoding information into inactive channels during the backward pass:

$$\mathbf{z}_{\text{input}} = \mathbf{M} \odot \mathbf{z} + (1-\mathbf{M}) \odot \text{sg}(\mathbf{z}), \tag{2}$$

where $\text{sg}(\cdot)$ denotes stop-gradient. This ensures gradients flow only through active channels, encouraging the encoder to prioritize information in early channels that are more likely to be retained. Alg. 1 formalizes the procedure.

Inference flexibility. At inference, we control compression rate by specifying active channel count $k \in [1, C]$. Given target k, we construct mask $\mathbf{M}$ deterministically with first k channels active, encode the image to obtain $\mathbf{z}$, and apply the mask. We quantize only the active channels via BSQ, while inactive channels are set to zero. The decoder reconstructs from both the quantized active channels and the zero-masked inactive channels, enabling continuous control over the rate-distortion tradeoff without retraining. The coarse-to-fine structure allows progressive decoding: reconstructions with $k = 32$ capture global structure, while increasing k progressively refines local details.

3.4 Binary Spherical Quantization

We incorporate Binary Spherical Quantization (BSQ) (Zhao et al., 2024) to discretize each active channel independently. Following the analysis in WeTok (Zhuang et al., 2025), BSQ provides a lookup-free, parameter-efficient alternative to codebook-based methods that is particularly well-suited for channel-wise tokenization. BSQ projects latent features onto a unit hypersphere and applies binary quantization. We use straight-through estimation (Bengio et al., 2013) for gradients and optimize entropy-based and commitment losses following (Zhao et al., 2024), with $\lambda_{\text{ent}} = 0.1$ and $\lambda_{\text{commit}} = 0.25$ in our quantization loss

$$\mathcal{L}_{\text{quant}} = \lambda_{\text{ent}}\mathcal{L}_{\text{ent}} + \lambda_{\text{commit}}\mathcal{L}_{\text{commit}}.$$

3.5 Training Objective

Reconstruction and perceptual losses. We supervise the decoder with an ℓ_1 reconstruction loss $\mathcal{L}_{\text{rec}} = \|\mathbf{x} - \hat{\mathbf{x}}\|_1$ and a VGG-based perceptual loss

$$\mathcal{L}_{\text{perc}} = \sum_l \|\Phi_l(\mathbf{x}) - \Phi_l(\hat{\mathbf{x}})\|_2^2, \tag{3}$$

where l indexes feature layers of the pretrained network Φ. The base objective combines these with the quantization loss:

$$\mathcal{L}_{\text{base}} = \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{perc}}\mathcal{L}_{\text{perc}} + \mathcal{L}_{\text{quant}}, \tag{4}$$

with fixed weights $\lambda_{\text{rec}} = 0.25$ and $\lambda_{\text{perc}} = 1.0$.

Table 1: **Flexible tokenization results on ImageNet-1K validation set.** We compare model size (M parameters), reconstruction quality, and encoder/decoder efficiency (throughput in images/s, latency in ms/image) at a 256-token budget. We additionally report a 512-token variant of our method. Best results are shown in **bold**, second-best are underlined.

Method	Params (M)			Reconstruction			Tput ↑		Latency ↓	
	Enc.	Dec.	Total	rFID ↓	LPIPS ↓	DreamSim ↓	Enc.	Dec.	Enc.	Dec.
OneDPiece	304	307	642	7.61	0.180	0.109	32.84	24.77	30.5	40.4
DOVE	70	**70**	287	13.82	**0.153**	0.133	**118.70**	**68.83**	**8.4**	**14.5**
ALIT	202	206	431	9.74	0.180	0.11	18.51	18.34	54.01	54.54
KARL	101	105	239	5.74	0.154	0.116	63.54	54.66	15.7	18.3
FlexTok	85	172	341	2.97	0.228	0.154	61.33	4.78	16.3	209.3
Ours (256)	**65**	95	**159**	3.70	0.169	0.111	51.34	41.39	19.5	24.2
Ours (512)	**65**	95	**159**	**2.92**	**0.153**	**0.096**	51.40	41.30	19.5	24.2

Adversarial training. To improve perceptual fidelity, we employ a PatchGAN discriminator (Isola et al., 2017) with the standard hinge loss (Miyato et al., 2018). We train for 15 epochs using only $\mathcal{L}_{\text{base}}$ as a warm-up, after which we add the adversarial term and optimize

$$\mathcal{L} = \mathcal{L}_{\text{base}} + \lambda_{\text{gan}}\mathcal{L}_{\text{hinge}}, \quad \lambda_{\text{gan}} = 0.1.$$

This schedule avoids early adversarial instability while boosting perceptual quality in later stages.

3.6 Training Considerations

Flexible-length channel-wise tokenization remains largely unexplored compared to spatial tokenization, with limited established training recipes in the literature. Through experimentation, we identify choices that impact training stability and reconstruction fidelity. We adopt $C = 512$ latent channels to balance capacity with stability. We employ ℓ_1 reconstruction loss with perceptual loss weighting ($\lambda_{\text{perc}} = 1.0$, $\lambda_{\text{rec}} = 0.25$), where higher perceptual weight enhances reconstruction fidelity; this reversal of traditional weighting aligns with recent findings in visual tokenization that prioritize perceptual quality (Bachmann et al., 2025; Yu et al., 2023). We also found ℓ_2 loss, Charbonnier loss, and gram matrix losses (Wu et al., 2024) produced noticeable blurring in our channel-wise setting, while perceptual weights below 0.1 caused training instabilities.

We introduce adversarial training to mitigate BSQ quantization artifacts, particularly checkerboard patterns and grid-like distortions that emerge from binary quantization. However, we delay GAN loss introduction until epoch 15, as applying it from initialization led to mode collapse in preliminary experiments. This approach allows the reconstruction objective to establish a stable baseline before adversarial refinement. Our stochastic masking ($p_{\text{mask}} = 0.5$) was more stable than deterministic warmup schemes that maintain full channels for initial epochs before transitioning to adaptive masking. Cosine decay with linear warmup (5K steps to peak LR 10^{-3}, decaying to 5×10^{-5}) becomes crucial when combining reconstruction and adversarial objectives (Brock et al., 2019; Karras et al., 2019). Finally, we adopt BSQ with an implicit codebook size of 2^{16} over codebook-based vector quantization since our aim was to first establish a training recipe for flexible channel-wise tokenization without additional complexity from codebook commitment and collapse issues (Yu et al., 2023). BSQ eliminates lookup operations and codebook memory storage, with recent work showing its relevance in channel-wise regimes (Zhuang et al., 2025).

4 Experiments

4.1 Flexible Length Tokenization

Evaluation protocol. We evaluate reconstruction quality and system efficiency on the 50k-image ImageNet-1K (Russakovsky et al., 2014) validation set using standard preprocessing. Reconstruction fidelity and perceptual quality are measured with PSNR, rFID (Jayasumana et al., 2024), L1 distance, LPIPS (Zhang et al., 2018), SSIM (Wang et al., 2004), and DreamSim (Fu et al., 2023). For system efficiency, we report encoding and decoding throughput in images per second, as well as latency

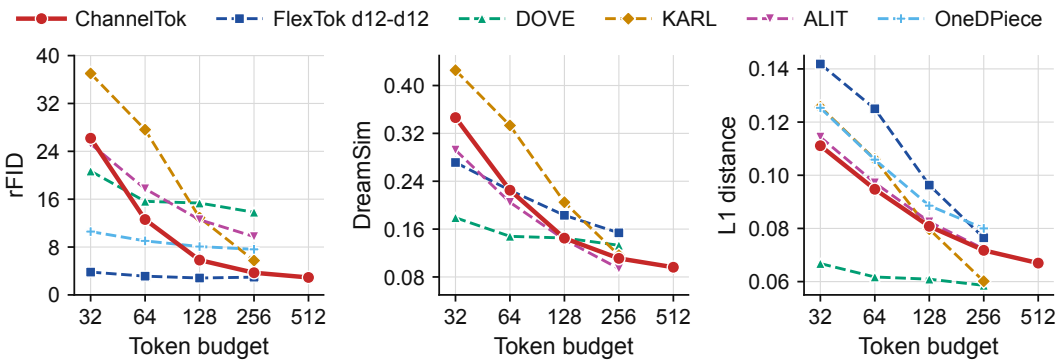

Figure 3: **Performance across token budgets.** Comparison of (a) rFID, (b) DreamSim, and (c) L1 distance as a function of token budget. ChannelTok consistently improves reconstruction quality with increasing token budgets while maintaining computational efficiency.

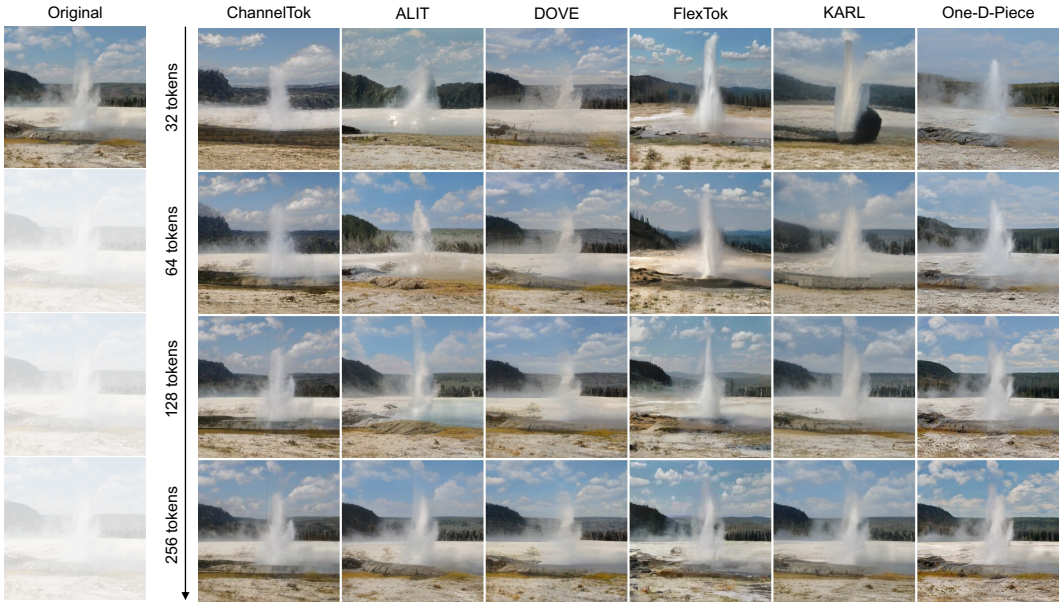

Figure 4: **Qualitative comparison across token budgets.** We show reconstructions from our model and prior flexible tokenizers at 32–256 tokens, along with the original image. Differences in sharpness, color consistency, and structural preservation can be observed as the token budget increases.

in milliseconds per image, measured on a single A5000 GPU. All methods are benchmarked with 128 batch size and FP16 precision, except DOVE, which requires BF16 due to FP16 incompatibility. Timing measurements include only the model forward pass and exclude data loading, GPU–CPU transfers, and disk I/O. All baselines are evaluated under the same protocol.

Performance at 256 tokens. Tab. 1 shows that our method establishes a new Pareto frontier for flexible tokenization, combining competitive perceptual quality with the smallest model size and high decoding throughput. We evaluate at a 256-token budget for fair comparison with key baselines (Bachmann et al., 2025; Miwa et al., 2025; Duggal et al., 2024) that support this maximum token count. At 256 tokens, our model achieves an rFID of 3.70, closely matching FlexTok's 2.97. At the same time, our decoder processes 41.4 images/s compared with 4.8 images/s for FlexTok, corresponding to an 8.6× speedup. Our full model contains 159M parameters versus FlexTok's 341M, reducing the parameter count by more than half. These efficiency gains stem from our lightweight architecture, which avoids the costly multi-step generative decoder used by FlexTok. DOVE achieves high throughput, but its rFID of 13.82 indicates considerably lower perceptual recinstruction quality.

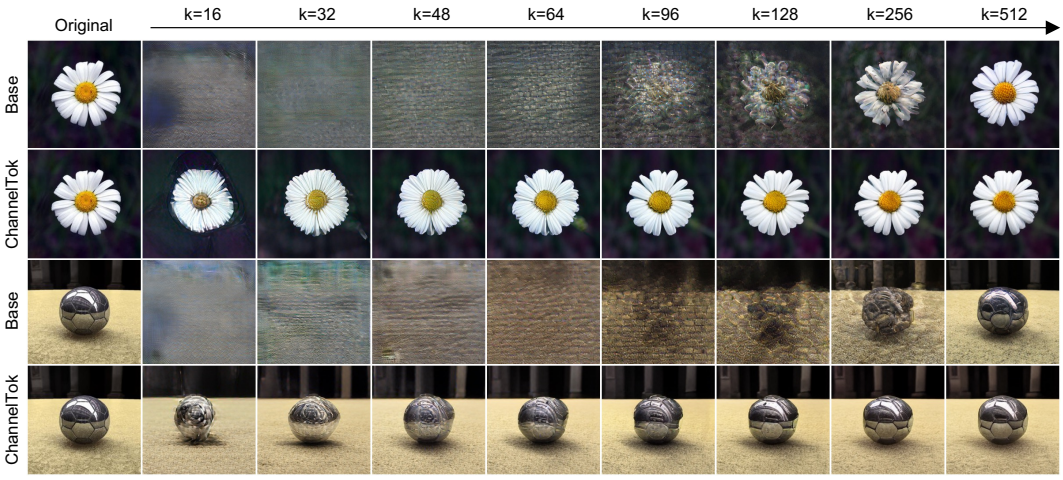

Figure 5: **Comparison of reconstructions with and without our channel-wise flexible masking.** The baseline model is identical to the flexible model but trained without masking. For the baseline: semantic coherence appears only at high token budgets. Our flexible tokenizer shows a semantic coarse-to-fine progression, with meaningful reconstructions even at low and moderate token budgets.

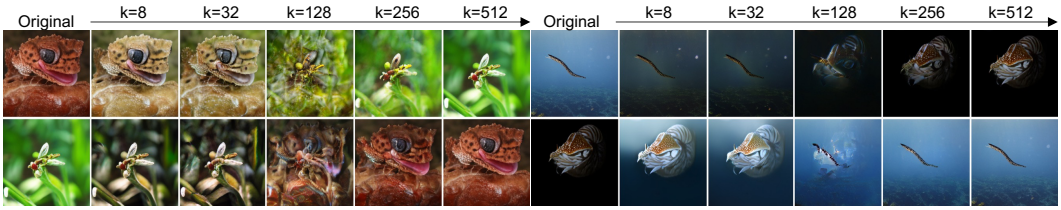

Figure 6: **Semantic organization in early channels.** Channel swapping experiments demonstrate hierarchical semantic encoding. Each pair of rows shows two images whose first k channels are progressively swapped. When channels from one image are replaced with another, the image progressively transforms from the source to the target.

Performance across token budgets. Fig. 3 shows that our channel-wise design scales with token capacity without retraining or architectural modifications. Reconstruction quality improves steadily across metrics as token capacity increases. At full 512-channel capacity, ChannelTok achieves the best rFID of 2.92 and DreamSim of 0.096 among the compared methods, while retaining the smallest model size at 159M parameters and one of the highest decoding throughputs. FlexTok maintains consistently low rFID across budgets, aided by its rectified-flow decoder (Bachmann et al., 2025), which provides perceptual coherence largely independent of token count. KARL and ALIT exhibit steeper rFID degradation at lower budgets but achieve strong pixel-level reconstruction at higher token counts. Together, our results show that ChannelTok supports a broad range of quality–efficiency operating points within a single model, with qualitative comparisons shown in Fig. 4.

4.2 Induced Semantic Hierarchy

Prefix Masking Enforces Ordering Fig. 5 illustrates the effect of our adaptive masking on the ordering of representations. Without tail-dropping, our baseline produces unstructured channels, with semantic content emerging only in the final tokens. Tail-dropping instead imposes a causal structure along the channel dimension, since tail channels may be masked during training. As a result, early channels learn to encode information independently and carry the most important content, while later channels add progressive refinement. Additional training details and qualitative examples appear in the supplementary material.

Semantic Transferability We investigate whether the global semantics encoded in early channels transfer between pairs of images. We select visually dissimilar image pairs and swap their first t

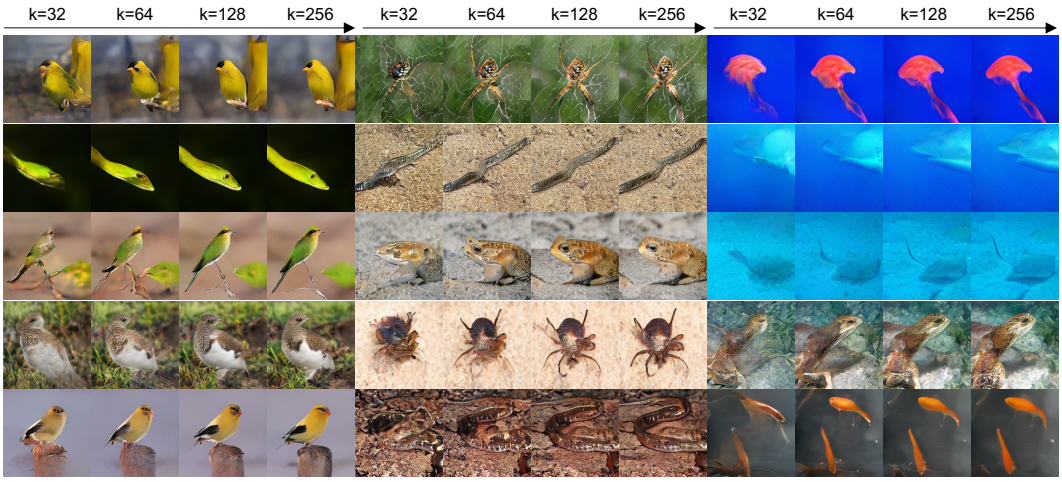

Figure 7: **Flexible autoregressive generation.** LlamaGen (Sun et al., 2024) GPT-L generates images channel by channel. At 32 tokens, samples preserve global structure and improve with more channels.

Table 2: **Token budget vs. speed/quality** (tokenizer reference).

Budget	Speedup	Time/Im.	gFID†
32	7.93×	0.475s	20.71
64	4.13×	0.92s	9.75
128	2.06×	1.82s	7.96
256	1.0×	3.78s	7.85

†FID w.r.t. tokenizer reconstruction.

Table 3: **Quantizer performance across token budgets.**

	FSQ		BSQ	
Budget	PSNR	L1	PSNR	L1
32	15.37	0.1166	15.58	0.1158
64	16.33	0.1020	16.42	0.1031
128	17.46	0.0881	17.34	0.0911
256	18.42	0.0775	18.04	0.0827

PSNR in dB; L1 is mean absolute error.

channels, as shown in Fig. 6. For small swaps ($t < 32$), we observe transfer of stylistic elements such as background characteristics, color temperature, and lighting while preserving the primary subject. As swap depth increases ($t \geq 64$), images undergo more noticeable semantic transformations with foreground subjects beginning to morph. This behavior mirrors the hierarchical refinement in progressive image coding, where early layers establish global structure before fine details emerge.

4.3 Autoregressive Image Generation

A key advantage of channel-wise tokenization is its natural compatibility with autoregressive generation. In spatial tokenization, each token encodes a local patch, and the generation order is an arbitrary raster scan, where truncating the sequence yields an incomplete spatial grid rather than a usable image. In contrast, our channel-wise tokens are ordered by semantic importance: each successive token refines the *entire* image, progressing from coarse structure to fine detail. This ordering maps directly onto the autoregressive factorization, making variable-length generation not merely possible but semantically meaningful.

We validate this property by training a LlamaGen (Sun et al., 2024) GPT-L transformer with 343M parameters on the discrete token sequences produced by our tokenizer. Tab. 2 reports FID relative to the tokenizer reconstructions and generation throughput across different token budgets on a single H100 GPU. With 64 tokens, the model achieves an FID of 9.75 while generating images 4.1 times faster than with the full 256-token sequence. Generation time decreases from 3.77 seconds per image at 256 tokens to 0.91 seconds at 64 tokens. Increasing the budget from 128 to 256 tokens yields only a marginal improvement in FID, from 7.96 to 7.85. This trend is consistent with the learned coarse-to-fine hierarchy, in which early channels capture most of the semantic content while later channels provide diminishing perceptual refinement. Even with only 32 tokens, generation is 7.9 times faster while still producing samples with coherent global structure, as shown in Fig. 7. These results confirm

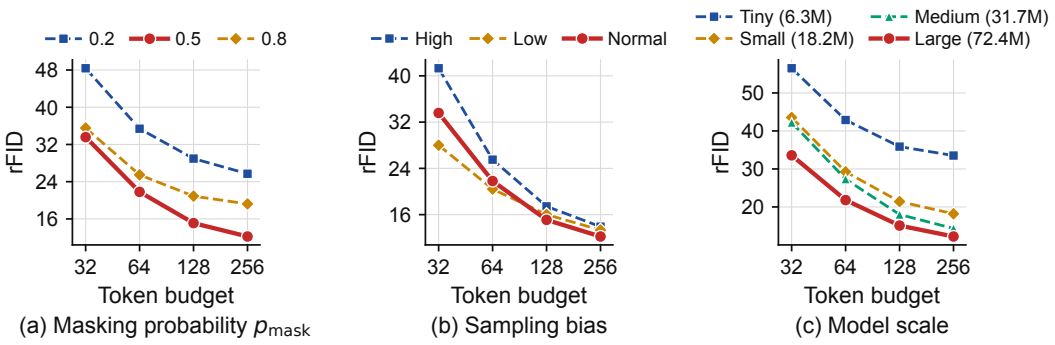

Figure 8: **Architectural ablations.** Effect of (a) masking probability p_{mask}, (b) sampling bias, and (c) model scale on rFID across token budgets. rFID improves consistently with more balanced masking, uniform sampling, and larger model capacity.

that the channel ordering learned by the tokenizer transfers naturally to autoregressive generation. Such a quality–speed trade-off is difficult to realize with spatial tokenizers, where truncating the token sequence typically yields spatially incomplete or fragmented images. In contrast, channel-wise generation allows a single trained model to operate across a wide range of latency budgets without retraining or architectural modification.

4.4 Ablation Studies

Effects of sampling and masking. As described in Sec. 3.3, our architecture introduces two forms of stochasticity: a global masking probability p_{mask} and a per-sample retention ratio t. We sweep p_{mask} on ImageNet-100 and measure rFID across token budgets. Low masking with $p_{\text{mask}} = 0.2$ performs worst as the model overfits to full-context reconstructions, while high masking with $p_{\text{mask}} = 0.8$ improves robustness but slightly hurts fidelity. We find $p_{\text{mask}} = 0.5$ optimal, providing sufficient masking pressure to induce semantic structure without destabilizing reconstructions. We then vary the retention distribution t at fixed $p_{\text{mask}} = 0.5$ in Fig. 8. As shown, biasing t towards lower or higher values trades off performance between token regimes. Uniform sampling provides the most balanced training signal across all budgets.

Scaling parameters. We train four models by varying the parameters of both the encoder and decoder. The trained models clearly exhibit a scaling curve, with larger models consistently improving rFID scores across token budgets as seen in Fig. 8.

Effect of quantizers. To study the effect of different quantizers we also train our tokenizer with FSQ on ImageNet-100 (Mentzer et al., 2023) and compare it with BSQ in Tab. 3. We observe that the channelwise tokenization is agnostic to the quantization method.

5 Conclusion

In this paper, we introduce a simple and efficient channel-wise flexible tokenizer that achieves strong reconstruction fidelity while maintaining high encoding and decoding throughput. By treating latent channels as visual tokens and applying stochastic prefix masking during training, our method learns an ordered coarse-to-fine representation in which early channels capture global semantic structure and later channels progressively refine visual details. This structure enables flexible compression across a wide range of token budgets without retraining or architectural modification, while avoiding the parameter-heavy backbones and multi-step generative decoders used by many prior flexible tokenizers. Experiments demonstrate that our approach achieves competitive reconstruction quality across token budgets while substantially improving model and decoding efficiency, and that the learned channel ordering transfers naturally to variable-length autoregressive image generation. Future work includes developing content-adaptive channel selection mechanisms, improving reconstruction quality in extremely low-token regimes, and extending channel-wise flexible representations to larger-scale image and video generation settings.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript drafting and editing, LaTeX formatting, and code cleanup. All AI-assisted outputs were reviewed by us authors, who take full responsibility for the final manuscript, code, claims, results, and associated artifacts.

REPRODUCIBILITY STATEMENT

Full training and evaluation details are provided in the appendix: architecture and optimization hyperparameters in Tab. 4 and Tab. 5, the loss configuration in Tab. 6, and the training-stability considerations that were necessary to reproduce our results in Sec. A.2. We have also shared our training recipe in Sec. 3.6. The evaluation protocol, including the exact metric implementations and preprocessing used for all reported numbers, is described in the appendix as well. All experiments use the publicly available ImageNet-1K and ImageNet-100 datasets with the standard transforms documented there.

REFERENCES

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. *arXiv preprint arXiv:2502.13967*, 2025.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. 2013.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *Int. Conf. Learn. Represent.*, 2019.

Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-Theory and Methods*, 3(1):1–27, 1974.

Shivam Duggal, Sanghyun Byun, William T Freeman, Antonio Torralba, and Phillip Isola. Adaptive length image tokenization via recurrent allocation. *arXiv preprint arXiv:2411.02393*, 2024.

Shivam Duggal, Sanghyun Byun, William T Freeman, Antonio Torralba, and Phillip Isola. Single-pass adaptive image tokenization for minimum program search. *arXiv preprint arXiv:2507.07995*, 2025.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12873–12883, 2021.

Stephanie Fu, Netanel Y Ramesh, Vongani H Xie, Yue Luo, Philip HS Torr, Joshua B Tenenbaum, Olga Russakovsky, William T Freeman, and Stephanie Wong. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1125–1134, 2017.

Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9307–9315, 2024.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 4401–4410, 2019.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham M. Kakade, Prateek Jain, and Ali Farhadi. Matryoshka representation learning. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, 2022.

Lingjun Mao, Zikang Jin, Haokui Wang, Xiaodan Zhang, and Xin Li. Images are worth variable length of representations. *arXiv preprint arXiv:2506.03643*, 2025.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505*, 2023.

Kazuki Miwa, Go Irie, Yuki Nakashima, and Rin-ichiro Taniguchi. One-d-piece: Image tokenizer meets quality-controllable compression. *arXiv preprint arXiv:2501.10064*, 2025.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1802.05957.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S Bernstein, Alexander C Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2014.

Junhong Shen, Kushal Tirumala, Michihiro Yasunaga, Ishan Misra, Luke Zettlemoyer, Lili Yu, and Chunting Zhou. Cat: Content-adaptive image tokenization. *arXiv preprint arXiv:2501.03120*, 2025.

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.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Detailflow: 1d coarse-to-fine autoregressive image generation via next-detail prediction. *arXiv preprint arXiv:2505.21473*, 2024.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Wentao Wu, Libin Huang, Wenyi Xu, Qi Chen, Yue Zhang, and Weiwei Zhou. AToken: Adaptive tokenization for vision transformers. *arXiv preprint arXiv:2509.14476*, 2024.

Wilson Yan, Matei Zaharia, Volodymyr Mnih, Pieter Abbeel, Aleksandra Faust, and Hao Liu. Elastictok: Adaptive tokenization for image and video. *arXiv preprint arXiv:2410.08368*, 2024.

Jingfeng Yao and Xinggang Wang. Quantize-then-rectify: Efficient vq-vae training. *arXiv preprint arXiv:2507.10547*, 2025.

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

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 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.

Yue Zhao, Yuanjun Panda, Zhengzhong Xu, Zhenzhong Wang, Gaurav Kumar, Yu Zhang, Jinshuo Zhou, Yan Chen, Guan Wang, Jiaqi Zhang, et al. Image and video tokenization with binary spherical quantization. *arXiv preprint arXiv:2406.07548*, 2024.

Shaobin Zhuang, Yiwei Guo, Canmiao Fu, Zhipeng Huang, Zeyue Tian, Fangyikang Wang, Ying Zhang, Chen Li, and Yali Wang. Wetok: Powerful discrete tokenization for high-fidelity visual reconstruction. *arXiv preprint arXiv:2508.05599*, 2025.