Channel-wise Vector Quantization: Native 1D Visual Tokenization with Learnable Ordering

Wei Song[1,2,3*] **Tianhang Wang**[1,3*] **Yitong Chen**[1,4] **Tong Zhang**[6]
Zuxuan Wu[1,4] **Min Li**[3] **Jiaqi Wang**[1,5†‡] **Kaicheng Yu**[2†]
[1]Shanghai Innovation Institute [2]Westlake University [3]Zhejiang University
[4]Fudan University [5]JD.COM [6]University of Chinese Academy of Sciences

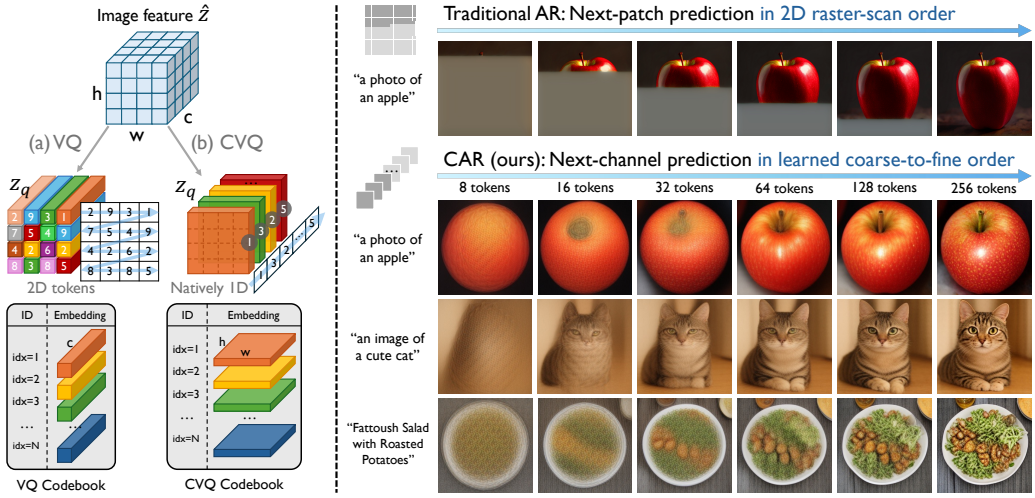


Figure 1: **(Left) VQ vs. CVQ.** Conventional VQ assigns an index to each $1\times1\times c$ patch feature vector, while CVQ assigns an index to each *channel* of the feature map. **(Right) AR vs. CAR.** Traditional autoregressive (AR) models generate images patch by patch in raster scan order (here, Emu3 [40]), whereas our channel-wise autoregressive (CAR) model generates images channel by channel in a 1D coarse-to-fine order. For example, given the prompt "a photo of an apple", the model first sketches a red circular shape corresponding to the apple's outline and dominant color, then progressively depicts its appearance, and finally adds fine-grained details such as specular highlights and yellow speckles.

Abstract

We present **Channel-wise Vector Quantization (CVQ)**, a novel image tokenization paradigm that replaces patch-wise tokens with channel-wise tokens. Unlike conventional vector quantization, which assigns a discrete token to each patch feature vector, CVQ quantizes each *channel* of the feature map. This represents an image as a 1D channel sequence with learnable coarse-to-fine ordering, rather than a 2D patch grid. Based on CVQ, we introduce a new visual autoregressive framework with ***"next-channel prediction"***. Instead of rendering images patch by patch in raster order, our **Channel-wise Auto-Regressive (CAR)** model predicts image channels sequentially, producing progressively enriched visual details. Specifically, it first sketches global structure and then refines fine-grained attributes, akin to a human artist's workflow. Empirically, we show that: (1) CVQ exhibits strong

*Co-first authors.
†Corresponding authors.
‡Project lead.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

codebook scalability, maintaining 95%+ utilization with up to 2^{16} entries while delivering growing reconstruction gains over VQ as the codebook scales up; and (2) CAR attains a DPG score of 86.7 and a GenEval score of 0.79, demonstrating strong effectiveness for text-to-image generation. The code is available at this link.

1 Introduction

Vector Quantization (VQ) [38, 10] is a fundamental technique for discretizing continuous image representations and serves as a cornerstone of discrete image generation [4, 33, 35]. However, since its introduction, the community has largely adhered to a patch-wise paradigm by default, where each index is assigned to represent a local $1 \times 1 \times c$ feature vector, as illustrated in Fig. 1 (left).

We argue that this long-standing convention imposes two major limitations. (1) *Insufficient codebook usage*, which leads to severe information loss and, consequently, poor reconstruction quality. While prior efforts have attempted to mitigate this issue [57, 58, 12], these works typically rely on complex tricks or extra parameters that increase structural complexity, or require token factorization [50, 33], which projects image features into a low-dimensional space for code index lookup. Such dimensionality reduction limits representational capacity [31] and substantially compromises token expressiveness [24]. (2) *Not naturally suited to sequence-to-sequence modeling*. Next-token prediction has achieved remarkable success in language models [1, 49], and the vision community seeks to mirror this success. However, language is inherently a 1D sequential signal (left-to-right), whereas images are spatial. Patch-wise tokenization discretizes images into 2D grids of tokens, which are then mechanically flattened into a 1D sequence to accommodate autoregressive (AR) learning (*e.g.*, via raster scan or z-curve). This structural mismatch results in a suboptimal token ordering for unidirectional AR modeling [18, 27, 53], as it disrupts local spatial dependencies among neighboring tokens [37, 17]. Furthermore, as discussed in [55], the strong local spatial bias of patch tokens makes it difficult to impose an AR-friendly ordering on them (*e.g.*, via nested dropout) [30].

In this paper, we show that *a simple change in the quantization axis* naturally resolves both limitations. Specifically, we shift VQ from a *patch-wise* to a *channel-wise* formulation. Unlike standard VQ, which learns a discrete *spatial* codebook of patch-wise indices, our CVQ learns a discrete *sequential* codebook of channel-wise indices.

Our motivation is intuitive: people typically draw by layering different levels of visual information to form a complete image. For example, when drawing an apple, an artist may first outline its overall shape and color tone before depicting finer details like the stem and speckles. Interestingly, as shown in Fig. 2, an autoencoder reflects a similar behavior by distributing different levels of visual information across channels, which jointly determine the final complete image.

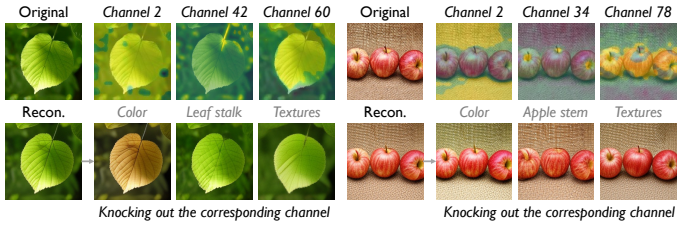


Figure 2: We visualize several channel activation maps from our VQ-VAE encoder and ablate individual channels by zeroing them before reconstruction. Interestingly, removing a channel selectively alters the corresponding content in the reconstructed image, with effects ranging from global appearance (*e.g.*, leaf color) to fine-grained structure (*e.g.*, removing the apple stem).

Motivated by this, unlike conventional VQ, which discretizes spatial vectors at each x,y location, we propose to discretize each feature channel. Since each channel tends to capture different aspects of visual information, CVQ naturally represents an image as a 1D token sequence with progressively enriched visual content. In contrast to conventional VQ, which imposes an artificial spatial ordering on the token sequence, CVQ internalizes spatial information within each token, as each channel encodes global spatial structure. As a result, the resulting token sequence forms a clean 1D representation, free from the limitations induced by patch-based tokenization.

On the other hand, as discussed in [57, 58], low codebook utilization in VQ arises from biased optimization dynamics, where only a small fraction of codebook vectors receive updates while the rest remain stagnant. We argue that this issue stems from the redundancy among image patches [15]. As analyzed in Fig. 5 (a), for two images with similar textures, patch-wise partitioning produces highly overlapping embeddings, causing a large number of patch-wise embeddings within each training

batch to be clustered to the same codebook indices (Fig. 5 (b)). Consequently, codebook updates concentrate on only a small subset of entries, leading to dead codes and codebook collapse (Fig. 5 (c)). In contrast, partitioning features along the channel dimension yields more separable embeddings across images, resulting in broader codebook coverage and improved codebook utilization.

In summary, our contributions are as follows:

- We provide a new perspective on vector quantization, showing that a simple change in the quantization axis effectively mitigates codebook collapse without additional modules or constraints.
- We propose a novel visual tokenization paradigm that represents images as native 1D sequences of channels, rather than 2D grids of patches, and show that, with simple nested dropout [30], the 1D channel tokens can learn a meaningful coarse-to-fine ordering that fits naturally with AR learning.
- Building on CVQ, we reformulate autoregressive image generation as a *next-channel prediction* framework that progressively enriches visual content from global structures to fine-grained details.

2 Related Work

Vector Quantization Vector quantization is essential for discretizing visual signals into tokens. VQ-VAE [38] first introduced a learnable codebook to obtain discrete latent representations. Building upon this, VQGAN [10] incorporates adversarial and perceptual losses to enhance image fidelity. RQ-VAE [22] and MoVQ [56] further reduce quantization error through multi-stage quantization and vector modulation. Despite these advances, low codebook utilization remains a central challenge in VQ. To mitigate this issue, ViT-VQGAN [50] proposes token factorization by projecting image features into a low-dimensional space for code index lookup. FSQ [26] and LFQ [51] extend this idea by quantizing representations into a small set of fixed values to prevent codebook collapse. However, such approaches substantially limit representational capacity [24, 31]. Recent works, including VQGAN-LC [57] and SimVQ [58], achieve higher codebook utilization through CLIP-feature initialization and learnable bases. However, these methods rely on more complex training pipelines and additional parameters compared to the original VQ architecture. IBQ [31] focuses on improving gradient propagation through index backpropagation and is orthogonal to our work. To the best of our knowledge, [48] is the only work that performs quantization along channels. However, it solely serves as an auxiliary component to support anytime sampling under computational constraints, and has not been systematically studied or evaluated as an independent vector quantization method. So far, dominant VQ methods remain patch-wise and thus inherit the limitations of 2D grid tokenization.

Autoregressive Visual Generation Inspired by the success of Large Language Models (LLMs) [3, 1], autoregressive (AR) image generation has become a current research hotspot. Existing methods typically tokenize images into 2D grids using VQGAN-like models and flatten them into raster-scan sequences, creating a structurally misaligned ordering for next-token prediction [33, 35, 40]. On the other hand, VAR [37] proposes next-scale prediction, which tokenizes images into multi-scale 2D tokens with bidirectional modeling within each scale. Infinity [13] further scales this approach to a much larger vocabulary size, achieving promising results. However, VAR deviates from the standard next-token prediction paradigm of LLMs, and its multi-scale hierarchy relies on heuristic partitioning.

Another line of work explores 1D visual tokenization. TiTok [52], SpectralAR [17] and Hita [55] resample 2D image representations into 1D sequences via learnable queries, while FlexTok [2] and Semanticist [41] extend the detokenizer to diffusion models. However, these methods rely on additional modules, such as token resampling modules and diffusion decoders [2, 41, 8], leading to complex architectures, potential information bottlenecks, and increased learning difficulty. In contrast, CVQ operates at a different conceptual level. As illustrated in Fig. 3, CVQ's 1D structure arises directly from the quantization process, enabling a natively 1D token sequence without specialized architectural design while maintaining the standard next-token prediction paradigm.

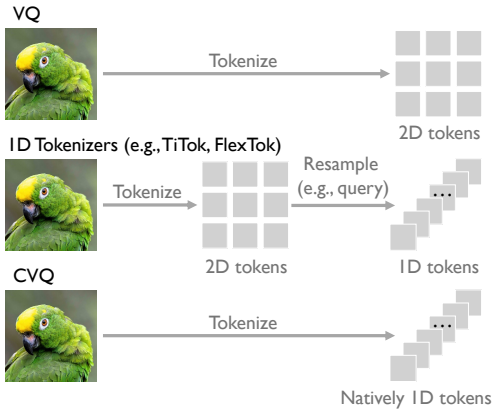


Figure 3: **Visual Tokenization Paradigms.**

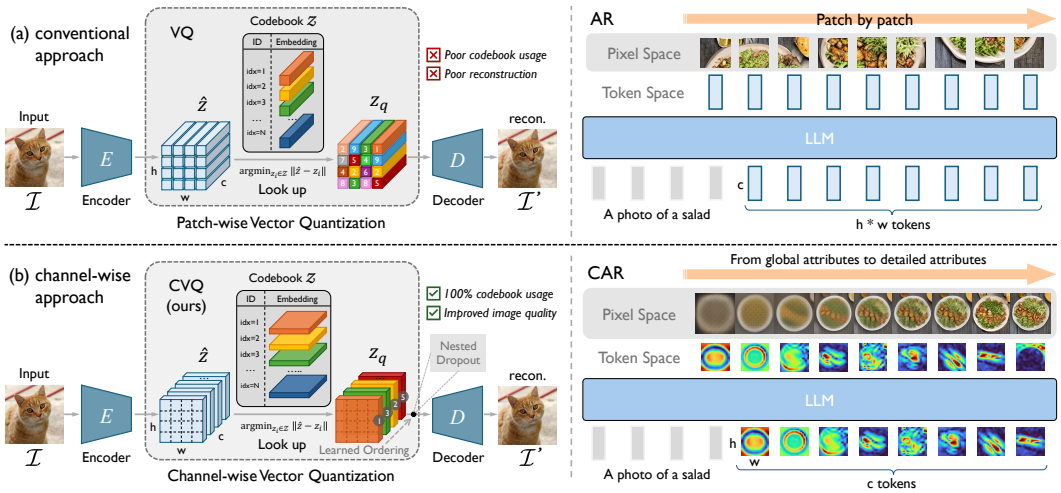


Figure 4: **(a) Conventional VQ and standard autoregressive image generation.** Conventional VQ assigns an index to each $1\times1\times c$ patch feature vector, which causes poor codebook usage and reconstruction quality. Furthermore, standard AR models predict tokens patch by patch, which imposes inherent sequential constraints on 2D spatial modeling. **(b) Proposed CVQ and CAR.** Our CVQ assigns an index to each $h \times w \times 1$ channel feature vector, enabling high codebook utilization, improved reconstruction quality, and an orderable 1D token space. Building on this, the proposed CAR model predicts visual content hierarchically, progressing from global structures to fine details.

3 Method

3.1 Channel-wise Vector Quantization

Traditional VQ frameworks map continuous latent variables to a discrete spatial codebook. As shown in Fig. 4(a), given an input image $\mathcal{I} \in \mathbb{R}^{H\times W\times 3}$, it is encoded into a latent representation $\mathbf{Z} \in \mathbb{R}^{h\times w\times c}$, where $h = H/f$, $w = W/f$ and f is the downsample ratio. In conventional VQ schemes, $\mathbf{Z}$ is treated as a collection of $M = h \times w$ spatial patch vectors, formulated as

$$\mathbf{Z} = \begin{bmatrix} \mathbf{z}^{(1,1,:)} & \mathbf{z}^{(1,2,:)} & \dots & \mathbf{z}^{(1,w,:)} \\ \mathbf{z}^{(2,1,:)} & \mathbf{z}^{(2,2,:)} & \dots & \mathbf{z}^{(2,w,:)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{z}^{(h,1,:)} & \mathbf{z}^{(h,2,:)} & \dots & \mathbf{z}^{(h,w,:)} \end{bmatrix}, \tag{1}$$

where $\mathbf{z}^{(i,j,:)} \in \mathbb{R}^{1\times1\times c}$ denotes a spatial vector at the position (i, j). The quantization process involves a nearest-neighbor lookup within the codebook $\mathcal{C}_{\text{spatial}} = \{\mathbf{e}_n\}_{n=1}^{N}$, where N denotes the codebook size:

$$\mathbf{z}_q^{(i,j,:)} = \underset{\mathbf{e}_n\in\mathcal{C}_{\text{spatial}}}{\operatorname{argmin}} \left\| \mathbf{z}^{(i,j,:)} - \mathbf{e}_n \right\|_2^2. \tag{2}$$

However, this conventional VQ paradigm typically suffers from codebook collapse, where only a small portion of the codebook receives gradient updates during training. [57, 58, 31].

We argue that this issue stems from the similarity and repetition between image patches [15]. As illustrated in Fig. 5(a), the patch-wise partitioned embeddings from different images are heavily entangled, resulting in substantial overlap of codebook indices both within and across images (Fig. 5(b)). Such redundancy and recurrence cause patches to cluster around the same vectors during early training. This, in turn, leads to the "death" of other cluster centers, ultimately resulting in codebook collapse (Fig. 5(c)).

In contrast to conventional VQ, CVQ introduces a distinct quantization mechanism, as shown in Fig. 4(b). The latent representation is partitioned as

$$\mathbf{Z} = \left[\mathbf{z}^{(:,:,1)}, \mathbf{z}^{(:,:,2)}, \dots, \mathbf{z}^{(:,:,c)}\right], \tag{3}$$

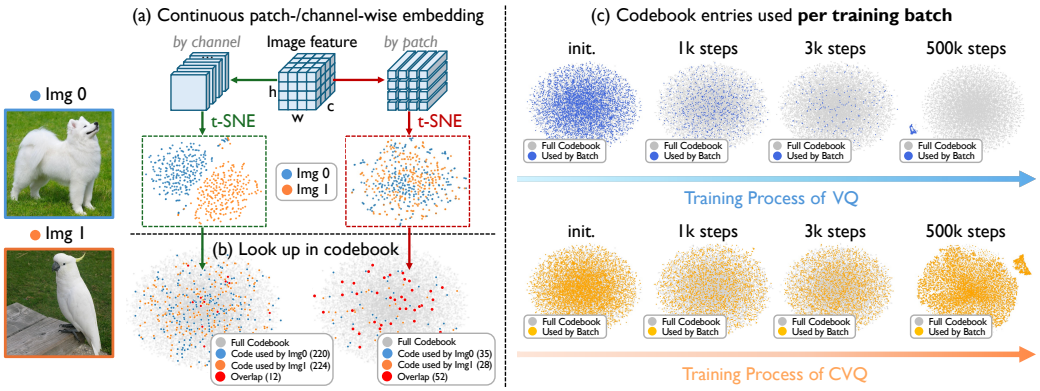


Figure 5: **(a) t-SNE visualization of embedding distributions.** For images with similar appearances but distinct semantics. Patch-wise partitioned embeddings of the two images are highly intermingled within a shared distribution, whereas channel-wise partitioning produces clearly separable clusters. **(b) t-SNE of codebook vectors used by each image.** Patch similarity leads to significant intra- and inter-image overlap across codebook entries. In contrast, channel-wise embeddings are distributed over a broader range of codebook points, resulting in higher utilization. More results are provided in Appendix. A. **(c) t-SNE of codebook entries used per training batch.** Conventional VQ rapidly collapses to a small subset of outlier entries, while CVQ maintains high codebook utilization throughout the training process.

where $\mathbf{z}^{(:,:,k)} \in \mathbb{R}^{h\times w\times 1}$ denotes the kth channel of $\mathbf{Z}$. For notational convenience, we denote $\mathbf{z}^{(:,:,k)}$ as $\mathbf{z}^{(k)}$ in the remainder of the paper. The quantization process is

$$\mathbf{z}_q^{(k)} = \underset{\mathbf{e}_n \in \mathcal{C}_{\text{channel}}}{\operatorname{argmin}} \left\| \mathbf{z}^{(k)} - \mathbf{e}_n \right\|_2^2, \tag{4}$$

where $\mathcal{C}_{\text{channel}}$ represents the channel-wise codebook comprising codewords $\mathbf{e}_n \in \mathbb{R}^{h\times w\times 1}$. For the lookup of two-dimensional representations, we adopt a simple formulation based on the Frobenius norm, which is equivalent to flattening the matrix into a vector and performing a nearest-neighbor lookup. The forward pass is defined as

$$\mathbf{z}_q^{(k)} = \mathbf{z}^{(k)} + \text{sg}[\mathbf{e}_n - \mathbf{z}^{(k)}], \tag{5}$$

where $\text{sg}[\cdot]$ denotes the stop-gradient operator. For the backward pass, CVQ utilizes the standard *Straight-Through Estimator* (STE) [38] approach, where gradients are copied directly from the quantized representation $\mathbf{z}_q$ to the continuous latent $\mathbf{z}$.

Within the CVQ framework, each codebook index represents a global channel feature rather than a local spatial patch. As illustrated in Fig. 5(a), partitioning representations along the channel dimension yields features that are highly distinguishable across different images. This property promotes high codebook utilization and reduces overlap, as evidenced in Fig. 5(b). Consequently, the t-SNE visualization in Fig. 5(c) shows that CVQ activates a significantly larger portion of the codebook within each training batch (batch size $= 64$) compared to standard VQ.

Tokenizer Training Following the standard VQGAN approach [10], the CVQ tokenizer is trained with a reconstruction objective comprising pixel-wise ℓ_2 loss, commitment loss, LPIPS loss [54], and adversarial loss with PatchGAN discriminator [19].

3.2 Channel-wise Autoregressive Generation

As illustrated in Fig. 4(b), we reformulate next-token prediction in autoregressive visual modeling by shifting from the traditional next-patch prediction paradigm to a *Next-Channel Prediction* (NCP) strategy, where the autoregressive unit is a *channel* rather than a patch token.

Through the CVQ tokenizer, an image is represented as a 1D sequence of c discrete channel tokens, $\mathcal{X} = \{x_i\}_{i=1}^{c}$. Subsequently, the decoder-only transformer is trained autoregressively to predict the next channel token conditioned on the textual context. The autoregressive likelihood of the entire

Table 1: **Reconstruction performance of different VQ methods on ImageNet-1K.** The vanilla form of CVQ improves reconstruction fidelity over VQ-based methods across both 256- and 1024-token budgets, outperforming established baselines such as MoVQ and VQ-LC at comparable codebook scales. Following prior work [57, 58, 31], we define codebook usage as the percentage of codebook entries selected at least once when tokenizing all 50,000 images in the ImageNet-1K validation set. † denotes models we reproduced using the same training setup as our CVQ tokenizer.

VQ METHODS	# Tokens	Ratio	Codebook Size	rFID ↓	SSIM↑	PSNR↑	Codebook Usage↑
256 tokens							
Vanilla VQ† [10]	16×16	16	16,384	4.84	0.542	19.93	4.5%
Dynamic VQ [16]	16×16	{8, 16, 32}	1,024	4.08	-	-	-
Dynamic VQ† [16]	16×16	{8, 16, 32}	16,384	3.62	0.558	20.07	5.2%
VQ-LC [57]	16×16	16	16,384	3.01	0.564	-	99.0%
VQ-LC [57]	16×16	16	100,000	<u>2.62</u>	**0.589**	-	99.0%
SimVQ† [58]	16×16	16	16,384	2.63	0.556	**21.10**	**100.0%**
Vanilla CVQ (ours)	256	16	16,384	**2.60**	<u>0.565</u>	<u>20.94</u>	**100.0%**
1024 tokens							
Vanilla VQ† [10]	32×32	8	16,384	1.32	0.692	23.35	2.8%
VQ w/ rotation trick [12]	32×32	8	16,384	1.10	-	-	27.0%
VQ-FC [50]	32×32	8	8,192	1.28	-	-	95.0%
VQ-LC [57]	32×32	8	100,000	1.29	<u>0.716</u>	-	99.5%
RVQ [22]	8×8×16	32	16,384	1.83	-	-	-
MoVQ [56]	16×16×4	16	1,024	1.12	0.673	22.42	-
MoVQ [56]	16×16×4	16	16,384	1.05	-	-	-
SimVQ† [58]	32×32	8	16,384	<u>0.90</u>	0.714	<u>24.88</u>	**100.0%**
Vanilla CVQ (ours)	1024	16	16,384	**0.88**	**0.723**	**25.02**	**100.0%**

image is thus given by

$$p(\mathcal{X}) = \prod_{k=1}^{c} p\Big(x^{(k)} \mid x^{(<k)}\Big). \tag{6}$$

To feed the channel tokens with dimension $h \times w$ into the transformer backbone, a two-layer MLP projector is applied to align their dimensionality with the LLM backbone.

Inducing Prefix-Based Channel Ordering Since channels do not possess an inherent order, we apply *nested channel dropout* during tokenizer training to induce a meaningful coarse-to-fine ordering for autoregressive modeling. Specifically, given a sequence of c channel tokens, we randomly sample a prefix length $c_{\text{keep}} \in \{1, \ldots, c\}$ and retain only the first c_{keep} channels, masking the remaining channels to zero. By enforcing reconstruction from variable-length channel prefixes, nested dropout induces a *coarse-to-fine* hierarchy, where early channels capture global structure and later channels progressively encode finer details [30]. This learned ordering enables progressive reconstruction as more channels are retained, as visualized in Fig. 6. We further analyze the effect of nested channel dropout in Sec. 4.3, with additional details provided in Appendix B.

4 Experiments

This section provides a comprehensive experimental analysis of CVQ's performance in downstream image reconstruction and generation tasks. In the controlled 256-token comparison, we set $c = h \times w = 256$, so CVQ and patch-wise VQ quantize the same latent tensor with identical token counts, codeword dimensions, and nearest-neighbor lookup complexity.

4.1 Visual Reconstruction

Training Setup We train two versions of our vision tokenizer with $c = 256$ and $c = 1024$ latent channels, yielding 256 / 1024 channel tokens, respectively. Both use a 16×16 latent grid, so each codeword has 256 dimensions. Unless otherwise specified, we use a codebook size of 16,384. All models are trained on ImageNet-1K [9] at 256×256 resolution for 100 epochs. We use Adam

Table 2: **Quantitative comparison on text-to-image generation benchmarks.** We categorize methods by attention mask into *Bidirectional*, *Unidirectional*, and *Hybrid* approaches. Bidirectional methods include diffusion and MaskGIT; unidirectional methods follow the standard AR paradigm of next-token prediction; while hybrid methods include the VAR family, which uses block-wise attention masks with full attention within each block. *NPP*, *NSP*, and *NCP* denote *next-patch*, *next-scale*, and *next-channel prediction*, respectively. Among them, only NPP and NCP follow the standard next-token prediction paradigm. † indicates results obtained with prompt rewriting or self-CoT.

Type	Methods	# Para.	GenEval↑				DPG↑		
			Two Obj.	Position	Color Attri.	**Overall**	Global	Relation	**Overall**
Bidirectional									
Diff.	SDXL [28]	2.6B	0.74	0.15	0.23	0.55	83.27	86.76	74.65
	SD3 (d=38) [11]	8B	0.89	0.34	0.47	0.71	-	-	-
	Lumina-Next [60]	1.7B	0.49	-	0.15	0.55	86.89	86.59	80.50
	FLUX-dev [21]	12B	-	-	-	0.67	-	-	84.00
	FLUX-schnell [21]	12B	-	-	-	0.71	-	-	84.80
	SANA [45]	1.6B	0.77	-	0.47	0.66	-	91.90	84.80
Mask.	TiTok [52]	0.6B	-	-	-	0.49	-	-	-
	TA-TiTok [20]	1.1B	0.58	0.13	0.34	0.55	-	-	-
Hybrid									
*NSP**	VARGPT-1.1 [59]	9B	0.53	0.13	0.21	0.53	84.83	88.13	78.59
	STAR [25]	1.7B	-	-	-	0.55	-	-	-
	TokenFLow [29]	7B	0.72	0.45	0.42	0.63	78.72	85.22	73.38
	HART [34]	0.7B	-	-	-	0.56	-	-	80.89
	SWITTI [39]	3B	-	-	-	0.62	-	-	-
	Infinity [13]	2B	0.85†	0.49†	0.57†	0.73†	93.11	90.76	83.46
	InfinityStar [23]	8B	0.90†	0.62†	0.67†	0.79†	91.68	91.87	86.55
	VAR [37]	4B	0.85	0.60	0.56	0.72	86.76	92.82	82.18
Unidirectional									
*NPP**	LlamaGen [33]	0.8B	0.34	0.07	0.04	0.32	-	-	64.84
	Chameleon [35]	7B	-	-	-	0.39	-	-	-
	Emu3 [40]	8B	0.81†	0.49†	0.45†	0.66†	-	-	80.60
	Lumina-mGPT [60]	7B	0.77	-	0.32	0.56	-	91.29	79.68
	Janus [42]	1.3B	0.68	0.46	0.42	0.61	-	-	-
	Liquid [43]	7B	0.73	0.17	0.37	0.55	-	-	-
	UniTok [24]	7B	0.71	0.26	0.45	0.59	-	-	-
	MUSE-VL [47]	7B	-	-	-	0.57†	-	-	-
	NextStep-1 [36]	14B	-	-	-	0.73†	-	-	85.28
	AR (4B)	4B	0.80	0.45	0.54	0.69	82.98	90.99	79.86
*NCP**	CAR (4B)	4B	0.86	0.62	0.59	0.74	86.93	93.77	82.14
	CAR (8B)	8B	0.92	0.66	0.66	0.79	89.40	94.16	86.72

($\beta_1 = 0.5$, $\beta_2 = 0.9$) with a learning rate of 1×10^{-4}, weight decay of 10^{-4}, and a global batch size of 256. We further extend CVQ to variable resolution in Appendix D.

Main Results We measured reconstruction FID (rFID), PSNR, and SSIM on the ImageNet-1K (val). As shown in Table 1, compared to conventional VQ, whose codebook utilization collapses to only 4.5% when scaling the codebook size to 16,384, CVQ natively supports 100% codebook utilization without requiring any additional modifications. This outperforms prior VQ improvement methods such as VQGAN-FC, which require additional token factorization, and VQGAN-LC, which relies on initializing the codebook using features extracted from a pretrained CLIP model and introduces extra projector parameters. Consequently, CVQ demonstrates a remarkable and consistent improvement over traditional VQ-based methods across different token budgets. With 256 tokens, CVQ achieves an rFID of 2.60, significantly improving over vanilla VQGAN (4.84) and outperforming SimVQ (2.63) in reconstruction fidelity. Under the 1024-token setting, the improvement becomes more pronounced: CVQ attains a lower rFID of 0.88 and a higher PSNR of 25.02 dB, surpassing strong baselines such as MoVQGAN (1.05 rFID) and VQGAN-LC (1.29 rFID).

4.2 Visual Generation

Training Recipe Our CAR models are initialized from the pre-trained Qwen3-4B/8B backbones [49]. The training process is conducted on 80M text-image pairs and is divided into two stages:

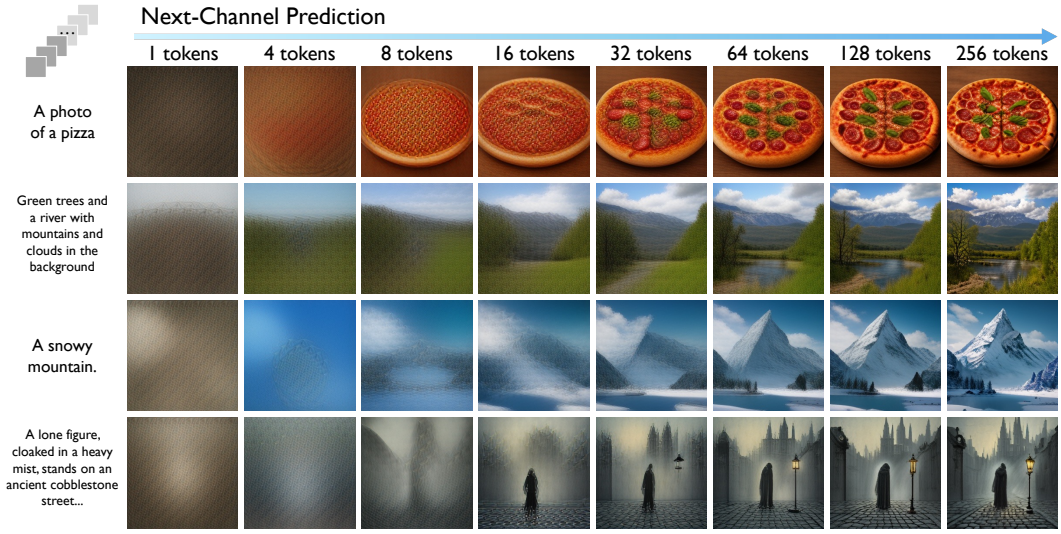


Figure 6: **The generation process of CAR.**

- **Stage I:** To align the CVQ features with the LLM latent space, we employ a 2-layer MLP projector. This projector maps the 256-dimensional channel embeddings to the hidden dimension of the LLM backbones (2560 for the 4B model and 4096 for the 8B model). During this stage, only the MLP projector and the LLM head are optimized, while the LLM backbone remains frozen.
- **Stage II:** In the second stage, we perform end-to-end optimization across all parameters, including the MLP projector, the LLM backbone, and the LLM head. We list the data sources and training hyperparameters in the Appendix.

Main Results As illustrated in Fig. 6, CAR produces progressively detailed image content as more channels are generated. As shown in Table 2, among unidirectional methods, CAR (4B) achieves competitive or superior performance compared with strong AR baselines such as NextStep-1 (14B) and Emu3 (8B). Scaling CAR to 8B further improves performance, reaching a GenEval score of 0.79 and a DPG overall score of 86.72, competitive with strong VAR methods such as Infinity and InfinityStar. These results suggest that CVQ provides an effective token ordering for AR learning while preserving a simple next-token prediction formulation. Qualitative results are shown in Fig. 7.

In addition to semantic-alignment benchmarks, we report MJHQ-30K FID in Table 6. CAR achieves an FID of 6.42, outperforming both 1D masked-token baselines and the standard 2D-token baseline.

4.3 Discussions

Codebook Scalability As shown in Table 3, scaling up the codebook size leads to a severe decline in utilization for patch-wise VQ, dropping from 20.3% to 1.1%, with only marginal improvements in rFID. In contrast, CVQ maintains nearly 100% utilization, even at massive scales (up to $65,536$). This enables CVQ to effectively leverage larger vocabularies for improved performance, reducing rFID from 3.64 to 2.32. Notably, such advantage becomes increasingly evident at scale: at a codebook size of 65K, CVQ achieves a 52% improvement in reconstruction fidelity over the VQ baseline, exhibiting promising scaling properties.

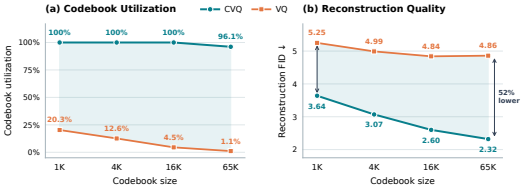


Figure 8: **Codebook Scalability.** (Left) CVQ maintains over 95% utilization even with 2^{16} entries, while VQ suffers from codebook collapse as the codebook size increases. (Right) CVQ consistently improves reconstruction quality as the codebook scales up, with growing gains over VQ.

Nested Dropout As shown in Table 4, CVQ without nested dropout exhibits weaker AR generation performance, as channel tokens do not possess an inherent order, whereas patch tokens still retain a natural spatial ordering despite suboptimal raster scanning. Yet, compared with 2D patch tokens, the 1D nature of channel-wise tokens makes it easy to impose a meaningful AR-friendly ordering.

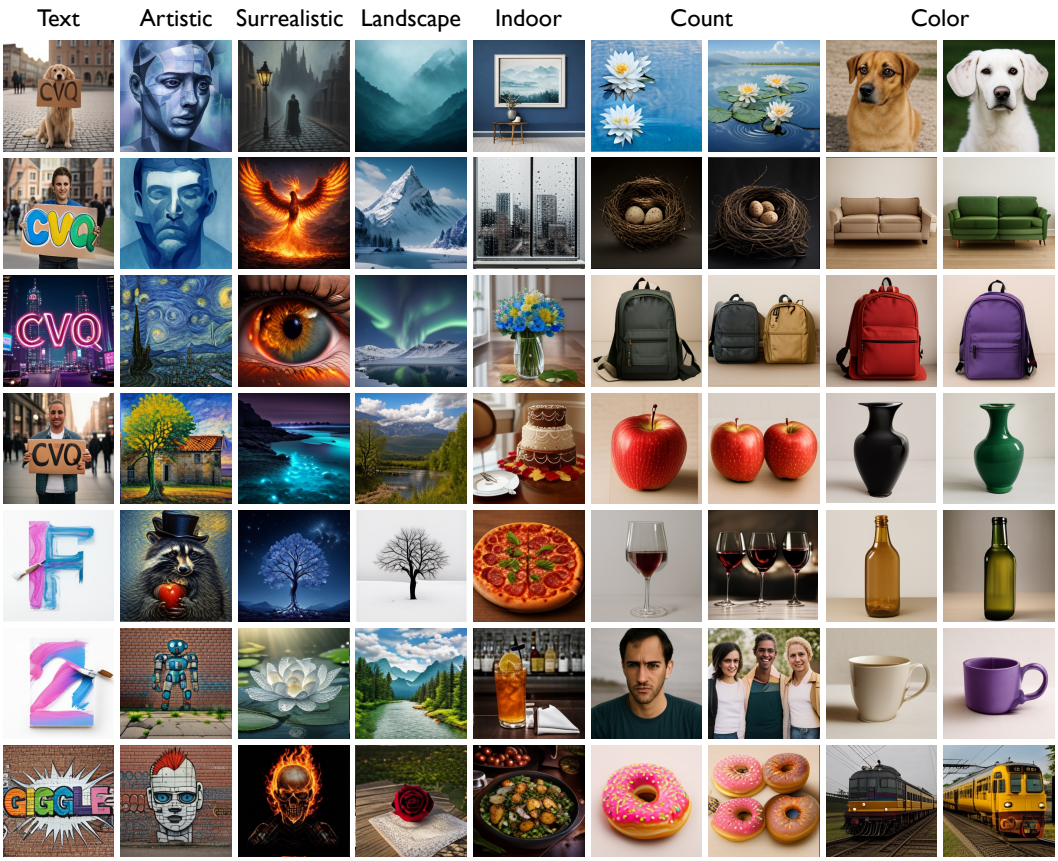


Figure 7: **Qualitative results on text-to-image generation.**

With a simple nested dropout strategy [30], CVQ learns a coarse-to-fine channel order, effectively improving AR performance while maintaining reconstruction quality. In contrast, such an ordering strategy is difficult to apply to 2D patch tokenization, as each patch token is strongly tied to local content and therefore cannot naturally represent a global coarse-to-fine progression.

Versus 1D Tokenizers While related in target, CVQ and existing 1D tokenizer works operate at different conceptual levels: the former is a quantization method, whereas the latter primarily focus on additional modifications at the model architecture level (*e.g.*, via learnable queries [52, 20, 17] or diffusion decoders [2, 41, 14]). CVQ's 1D structure arises directly from the quantization process itself, enabling a 1D token sequence without specialized model architectural design.

Here, we provide a direct comparison with recent 1D tokenizers. Since these methods typically employ stronger training recipes, such as two-stage training, external proxy codes, or enhanced networks for LPIPS and GAN losses, we report results under both the standard VQGAN-style training protocol and a stronger TA-TiTok-style recipe in Table 5.

Table 3: **Ablation on codebook usage across different codebook scales.** Conventional VQ suffers from increasingly severe codebook collapse as the vocabulary grows, whereas CVQ maintains high utilization even with large codebooks, demonstrating superior codebook scalability.

Codebook Size	VQ		CVQ	
	Util.↑	rFID ↓	Util.↑	rFID ↓
1,024	20.3%	5.25	100% (+79.7)	3.64(↓31%)
4,096	12.6%	4.99	100% (+87.4)	3.07(↓38%)
16,384	4.5%	4.84	100% (+95.5)	2.60(↓46%)
65,536	1.1%	4.86	96.1% (+95.0)	2.32(↓52%)

Table 4: **Ablation on nested dropout.** (1) Standard patch-wise VQ is incompatible with nested dropout; (2) nested channel dropout effectively induces an AR-friendly ordering over channel tokens, substantially improving generation performance (GenEval +0.12, DPG +9.38) without compromising reconstruction quality.

Method	Recon.			Generation	
	rFID ↓	SSIM↑	PSNR↑	GenEval↑	DPG↑
VQ w/o dropout	4.84	0.542	19.93	0.69	79.86
VQ w/ dropout	15.32	0.322	3.43	-	-
CVQ w/o dropout	**2.60**	0.565	20.94	0.62	72.76
CVQ w/ dropout	2.62	**0.568**	**21.01**	**0.74** (+0.12)	**82.14** (+9.38)

Table 5: **Comparison with representative 1D tokenizers.** †: standard VQGAN-style recipe, ‡: two-stage recipe with decoder reinforcement [52], ∗: models reproduced with the official codebase.

Method	Diff. Dec.	rFID ↓	SSIM ↑	PSNR ↑
FlexTok [2]	✓	4.20	-	-
FlexTok-1.4B [2]	✓	1.45	0.465	18.53
SpectralAR‡ [17]	✗	4.03	-	-
TiTok-64† [52]	✗	5.15	-	-
TiTok-256†∗ [52]	✗	3.84	0.542	19.92
CVQ†	✗	2.60	0.565	20.94
Same enhanced training recipe [20]				
TA-TiTok-64 [20]	✗	2.43	-	-
TA-TiTok-128 [20]	✗	1.53	-	-
TiTok-64 [52]	✗	4.25	-	-
TiTok-128 [52]	✗	2.63	-	-
TiTok-256∗ [52]	✗	1.51	0.558	20.05
CVQ	✗	**1.29**	**0.569**	**20.97**

Table 6: **FID performance on MJHQ-30K.** We further evaluate text-to-image generation fidelity across different methods on MJHQ-30K.

Method	Type	1D Token	Res.	FID ↓
SD-XL [28]	Diff.	✗	1024	9.55
PixArt [7]	Diff.	✗	512	6.14
FlowTok-H [14]	Diff.	✓	256	7.15
Show-o [46]	Discrete Diff.	✗	256	14.99
TiTok [52]	Mask.	✓	256	8.50
TA-TiTok [20]	Mask.	✓	256	7.51
STAR [25]	NSP	✗	256	5.67
SWITTI [39]	NSP	✗	512	9.50
SWITTI [39]	NSP	✗	1024	8.10
LlamaGen [33]	NTP	✗	512	25.59
VILA-U [44]	NTP	✗	256	12.81
Janus [42]	NTP	✗	256	10.10
UniTok [24]	NTP	✗	256	7.46
DualToken [32]	NTP	✗	256	7.88
CAR	NTP	✓	256	**6.42**

5 Conclusion and Future Work

Conclusion In this work, we introduced CVQ, a simple yet effective vector quantization paradigm that discretizes images along the channel dimension. CVQ achieves high codebook utilization at scale and strong reconstruction fidelity without auxiliary designs. Building upon CVQ, we proposed CAR, an autoregressive image generation framework that shifts the next-token prediction unit from patches to channels. Our results highlight channel-wise tokens as a promising direction for autoregressive image generation and offer new insight into rethinking the fundamental unit of visual tokenization.

Limitations CAR exhibits a distinct failure pattern compared with patch-wise AR. As each channel token spans the entire spatial field, severe channel-prediction errors can affect large image regions, leading to structural distortions or global blurring even when the overall semantics are preserved. In contrast, patch-wise AR failures more often manifest as semantic omissions or localized distortions.

Future Work Despite the encouraging results, several important directions remain for future work. First, CVQ can be combined with recent advances in VQ, such as SimVQ [58] and IBQ [31], to further improve representational capacity. Second, the autoregressive formulation of CAR makes it a natural fit for unified vision models that jointly perform visual understanding and generation within a single architecture. Finally, extending the channel-wise quantization paradigm to the temporal dimension represents a promising direction for learning efficient and compact video representations.

Acknowledgments

This work received computational support from SII and JD.COM and was partially supported by the Zhejiang Leading Innovative and Entrepreneurial Team Introduction Program (2024R01007). The authors declare no competing interests.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2, 3

[2] 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. URL https://openreview.net/forum?id=DgdOkUUBzf. 3, 9, 10, 16

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 2020. 3

[4] 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*, pp. 11315–11325, 2022. 2

[5] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021. 18

[6] Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025. 18

[7] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-alpha: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023. 10

[8] Yitong Chen, Zuxuan Wu, Xipeng Qiu, and Yu-Gang Jiang. Catok: Taming mean flows for one-dimensional causal image tokenization. In *CVPR*, 2026. 3

[9] 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*, pp. 248–255. Ieee, 2009. 6

[10] 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*, pp. 12873–12883, 2021. 2, 3, 5, 6, 18

[11] 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. 7

[12] Christopher Fifty, Ronald G Junkins, Dennis Duan, Aniketh Iyengar, Jerry W Liu, Ehsan Amid, Sebastian Thrun, and Christopher Ré. Restructuring vector quantization with the rotation trick. *The Thirteenth International Conference on Learning Representations*, 2025. 2, 6

[13] Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15733–15744, 2025. 3, 7

[14] Ju He, Qihang Yu, Qihao Liu, and Liang-Chieh Chen. Flowtok: Flowing seamlessly across text and image tokens. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16629–16640, 2025. 9, 10

[15] Kaiming He and Jian Sun. Statistics of patch offsets for image completion. In *European conference on computer vision*, pp. 16–29. Springer, 2012. 2, 4

[16] Mengqi Huang, Zhendong Mao, Zhuowei Chen, and Yongdong Zhang. Towards accurate image coding: Improved autoregressive image generation with dynamic vector quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22596–22605, June 2023. 6

[17] Yuanhui Huang, Weiliang Chen, Wenzhao Zheng, Yueqi Duan, Jie Zhou, and Jiwen Lu. Spectralar: Spectral autoregressive visual generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. 2, 3, 9, 10

[18] Zhihao Huang, Xi Qiu, Yukuo Ma, Yifu Zhou, Junjie Chen, Hongyuan Zhang, Chi Zhang, and Xuelong Li. Nfig: multi-scale autoregressive image generation via frequency ordering. *arXiv preprint arXiv:2503.07076*, 2025. 2

[19] 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. 5

[20] 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. 7, 9, 10

[21] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024. 7

[22] 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*, pp. 11523–11532, 2022. 3, 6

[23] Jinlai Liu, Jian Han, Bin Yan, Hui Wu, Fengda Zhu, Xing Wang, Yi Jiang, Bingyue Peng, and Zehuan Yuan. Infinitystar: Unified spacetime autoregressive modeling for visual generation. *arXiv preprint arXiv:2511.04675*, 2025. 7

[24] 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, 3, 7, 10

[25] Xiaoxiao Ma, Mohan Zhou, Tao Liang, Yalong Bai, Tiejun Zhao, Huaian Chen, and Yi Jin. Star: Scale-wise text-to-image generation via auto-regressive representations. *arXiv preprint arXiv:2406.10797*, 2024. 7, 10

[26] Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=8ishA3LxN8. 3

[27] Ziqi Pang, Tianyuan Zhang, Fujun Luan, Yunze Man, Hao Tan, Kai Zhang, William T Freeman, and Yu-Xiong Wang. Randar: Decoder-only autoregressive visual generation in random orders. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 45–55, 2025. 2

[28] 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. *The Twelfth International Conference on Learning Representations*, 2024. 7, 10

[29] Liao Qu, Huichao Zhang, Yiheng Liu, Xu Wang, Yi Jiang, Yiming Gao, Hu Ye, Daniel K Du, Zehuan Yuan, and Xinglong Wu. Tokenflow: Unified image tokenizer for multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025. 7

[30] Oren Rippel, Michael Gelbart, and Ryan Adams. Learning ordered representations with nested dropout. In *International Conference on Machine Learning*, pp. 1746–1754. PMLR, 2014. 2, 3, 6, 9, 16

[31] 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*, 2025. 2, 3, 4, 6, 10

[32] Wei Song, Yuran Wang, Zijia Song, Yadong Li, Zenan Zhou, Long Chen, Jianhua Xu, Jiaqi Wang, and Kaicheng Yu. Dualtoken: Towards unifying visual understanding and generation with dual visual vocabularies. *arXiv preprint arXiv:2503.14324*, 2025. 10

[33] 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, 3, 7, 10

[34] Haotian Tang, Yecheng Wu, Shang Yang, Enze Xie, Junsong Chen, Junyu Chen, Zhuoyang Zhang, Han Cai, Yao Lu, and Song Han. Hart: Efficient visual generation with hybrid autoregressive transformer. *The Thirteenth International Conference on Learning Representations*, 2025. 7

[35] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024. 2, 3, 7

[36] NextStep Team, Chunrui Han, Guopeng Li, Jingwei Wu, Quan Sun, Yan Cai, Yuang Peng, Zheng Ge, Deyu Zhou, Haomiao Tang, Hongyu Zhou, Kenkun Liu, Ailin Huang, Bin Wang, Changxin Miao, Deshan Sun, En Yu, Fukun Yin, Gang Yu, Hao Nie, Haoran Lv, Hanpeng Hu, Jia Wang, Jian Zhou, Jianjian Sun, Kaijun Tan, Kang An, Kangheng Lin, Liang Zhao, Mei Chen, Peng Xing, Rui Wang, Shiyu Liu, Shutao Xia, Tianhao You, Wei Ji, Xianfang Zeng, Xin Han, Xuelin Zhang, Yana Wei, Yanming Xu, Yimin Jiang, Yingming Wang, Yu Zhou, Yucheng Han, Ziyang Meng, Binxing Jiao, Daxin Jiang, Xiangyu Zhang, and Yibo Zhu. Nextstep-1: Toward autoregressive image generation with continuous tokens at scale. *arXiv preprint arXiv:2508.10711*, 2025. 7

[37] 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*, 2024. 2, 3, 7

[38] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, 2017. 2, 3, 5

[39] Anton Voronov, Denis Kuznedelev, Mikhail Khoroshikh, Valentin Khrulkov, and Dmitry Baranchuk. Switti: Designing scale-wise transformers for text-to-image synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025. 7, 10

[40] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024. 1, 3, 7

[41] Xin Wen, Bingchen Zhao, Ismail Elezi, Jiankang Deng, and Xiaojuan Qi. "Principal components" enable a new language of images. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. 3, 9, 16

[42] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *arXiv preprint arXiv:2410.13848*, 2024. 7, 10

[43] Junfeng Wu, Yi Jiang, Chuofan Ma, Yuliang Liu, Hengshuang Zhao, Zehuan Yuan, Song Bai, and Xiang Bai. Liquid: Language models are scalable and unified multi-modal generators. *International Journal of Computer Vision*, 2026. 7

[44] 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. *The Thirteenth International Conference on Learning Representations*, 2025. 10

[45] Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *The Thirteenth International Conference on Learning Representations*, 2025. 7

[46] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024. 10

[47] Rongchang Xie, Chen Du, Ping Song, and Chang Liu. Muse-vl: Modeling unified vlm through semantic discrete encoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 24135–24146, 2025. 7

[48] Yilun Xu, Yang Song, Sahaj Garg, Linyuan Gong, Rui Shu, Aditya Grover, and Stefano Ermon. Anytime sampling for autoregressive models via ordered autoencoding. *arXiv preprint arXiv:2102.11495*, 2021. 3

[49] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 2, 7

[50] 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. In *The Tenth International Conference on Learning Representations*, 2022. 2, 3, 6

[51] Lijun Yu, Jose Lezama, Nitesh Bharadwaj Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A Ross, and Lu Jiang. Language model beats diffusion - tokenizer is key to visual generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gzqrANCF4g. 3

[52] 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*, 2024. 3, 7, 9, 10

[53] 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*, pp. 18431–18441, 2025. 2

[54] 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. 5

[55] Anlin Zheng, Haochen Wang, Yucheng Zhao, Weipeng Deng, Tiancai Wang, Xiangyu Zhang, and Xiaojuan Qi. Holistic tokenizer for autoregressive image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16916–16926, October 2025. 2, 3

[56] Chuanxia Zheng, Tung-Long Vuong, Jianfei Cai, and Dinh Phung. Movq: Modulating quantized vectors for high-fidelity image generation. *Advances in Neural Information Processing Systems*, 35:23412–23425, 2022. 3, 6

[57] Lei Zhu, Fangyun Wei, Yanye Lu, and Dong Chen. Scaling the codebook size of vq-gan to 100,000 with a utilization rate of 99%. In *Advances in Neural Information Processing Systems*, 2024. 2, 3, 4, 6

[58] Yongxin Zhu, Bocheng Li, Yifei Xin, Zhihua Xia, and Linli Xu. Addressing representation collapse in vector quantized models with one linear layer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. 2, 3, 4, 6, 10

[59] Xianwei Zhuang, Yuxin Xie, Yufan Deng, Dongchao Yang, Liming Liang, Jinghan Ru, Yuguo Yin, and Yuexian Zou. Vargpt-v1.1: Improve visual autoregressive large unified model via iterative instruction tuning and reinforcement learning. *arXiv preprint arXiv:2504.02949*, 2025. 7

[60] Le Zhuo, Ruoyi Du, Han Xiao, Yangguang Li, Dongyang Liu, Rongjie Huang, Wenze Liu, Xiangyang Zhu, Fu-Yun Wang, Zhanyu Ma, et al. Lumina-next: Making lumina-t2x stronger and faster with next-dit. *Advances in Neural Information Processing Systems*, 2024. 7

A More t-SNE Visualizations of Embedding Distributions.

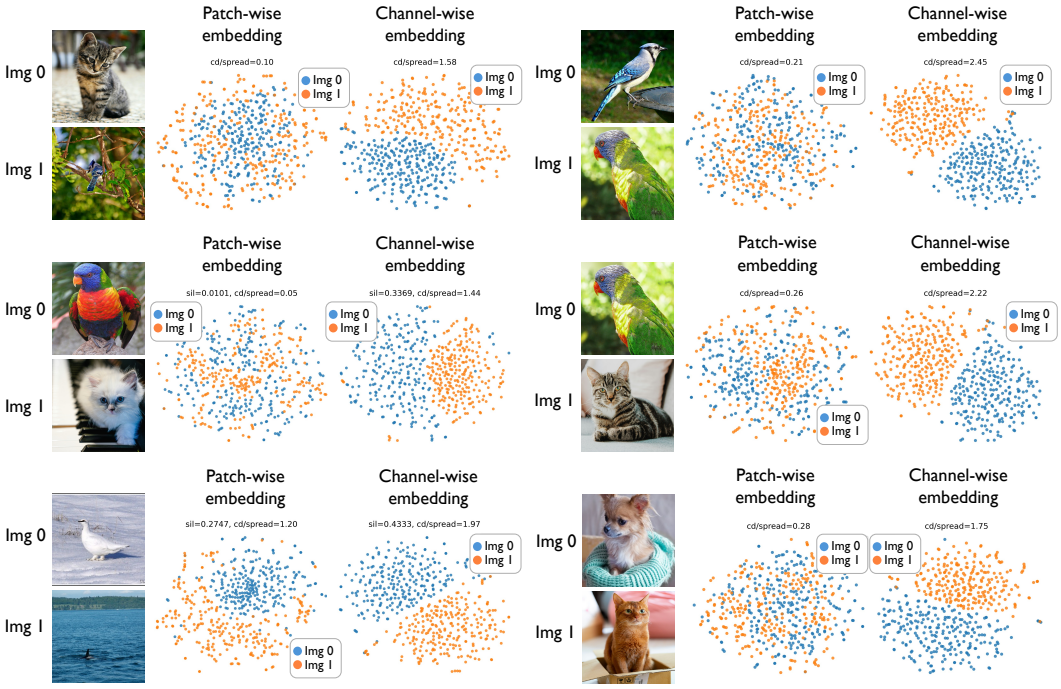


Figure 9: More visualizations of feature map channels

B Details and Analysis of the Nested Channel Dropout Strategy

B.1 Implementation Details of Nested Channel Dropout

During the training phase, we introduce a structured dropout mechanism to enforce a coarse-to-fine hierarchy within the latent space. For a given latent representation $\mathbf{Z} \in \mathbb{R}^{h \times w \times c}$, we define a truncated configuration where only the first c_{keep} channels are retained, while the remaining $c - c_{\text{keep}}$ channels are masked (set to zero). This truncated latent, denoted as $\mathbf{Z}_{\perp c_{\text{keep}}}$, effectively forces the model to compress the most critical visual information into the lower-indexed channels.

The optimization objective for a specific channel configuration c_{keep} is formulated as follows:

$$\mathcal{L}_{\text{nested}}(c_{\text{keep}}) = \mathcal{L}_{\text{recon}}(\mathbf{Z}_{\perp c_{\text{keep}}}) + \mathcal{L}_{\text{quant}}(\mathbf{Z}_{\perp c_{\text{keep}}}) + \mathcal{L}_{\text{lpips}}(\mathbf{Z}_{\perp c_{\text{keep}}}) + \lambda_{\text{GAN}}(c_{\text{keep}})\mathcal{L}_{\text{GAN}}(\mathbf{Z}_{\perp c_{\text{keep}}}), \quad (7)$$

where $\mathcal{L}_{\text{recon}}$ and $\mathcal{L}_{\text{lpips}}$ denote the standard reconstruction and perceptual losses, respectively. The quantization loss $\mathcal{L}_{\text{quant}}$ is computed exclusively over the active c_{keep} channels to ensure valid codebook mapping at reduced dimensions.

A key challenge in training nested representations is that the discriminator often suffers from instability when operating on extremely sparse latent features, which can lead to unnatural artifacts in reconstructions using few channels. To mitigate this, we introduce an adaptive GAN weight λ_{GAN} governed by a sigmoid function:

$$\lambda_{\text{GAN}}(c_{\text{keep}}) = \frac{\lambda_0}{1 + e^{-\eta(c_{\text{keep}} - \frac{c}{2})}}, \quad (8)$$

where $\eta = 0.05$ controls the transition smoothness, and $\lambda_0 = 1$ is the base GAN weight. This formulation ensures that the adversarial influence is gradually introduced as the channel capacity increases, preserving stable convergence for sparse configurations.

To balance hierarchical representation learning with overall reconstruction fidelity, we adopt a stochastic training strategy. In each iteration, nested channel dropout is performed with probability α (the dropout ratio), where the number of active channels c_{keep} is sampled from a discrete uniform

Table 7: **Ablation on nested channel dropout ratio.** We evaluate the impact of varying the dropout ratio α on reconstruction fidelity and downstream text-to-image generation. While reconstruction performance remain stable across different ratios, generation performance improves significantly with the introduction of channel dropout, with comparable gains observed at $\alpha = 25\%$ and $\alpha = 50\%$.

α	Recon.			Generation	
	rFID $^{\downarrow}$	SSIM$^{\uparrow}$	PSNR$^{\uparrow}$	Geneval$^{\uparrow}$	DPG$^{\uparrow}$
0%	2.60	0.565	20.94	0.62	72.76
25%	2.62	0.568	21.01	0.74 (+0.12)	82.14 (+9.38)
50%	2.61	0.562	20.88	0.75 (+0.13)	81.96 (+9.20)

distribution $\mathcal{U}(1, c)$. Conversely, with probability $1 - \alpha$, the model undergoes standard training using the full channel configuration. The resulting hybrid optimization objective is formulated as:

$$\mathcal{L}_{\text{sum}} = \alpha \cdot \mathbb{E}_{c_{\text{keep}} \sim \mathcal{U}(1,c)}[\mathcal{L}_{\text{nested}}(c_{\text{keep}})] + (1 - \alpha)\mathcal{L}_{\text{total}}, \tag{9}$$

where $\mathcal{L}_{\text{total}}$ represents the loss under the complete channel configuration ($c_{\text{keep}} = c$).

B.2 Analysis of Nested Channel Dropout

[30] demonstrates that a nested dropout strategy, which stochastically removes nested sets of hidden units, enforces an ordered representation where importance decreases with the dimension index. For semi-linear autoencoders (comprising a linear or sigmoidal encoder with a linear decoder) under L_2 loss, they establish that: (i) its optimal solutions are a subset of standard autoencoder optima, preserving global performance (Theorem 1); (ii) the weight matrices must be commutative in their truncation and inversion (Theorem 2); and (iii) with an orthonormality constraint ($\Gamma^T\Gamma = \mathbb{I}_K$), the model uniquely recovers the PCA solution (Theorem 3).

However, these theoretical guarantees necessitate stringent conditions: the architecture must be semi-linear and optimized via L_2 reconstruction loss. While the proposed unit sweeping technique fixes converged units to mitigate gradient decay at higher indices, its iterative nature remains computationally expensive for modern deep learning.

Nevertheless, empirical evidence [2, 41] suggests that even with deeper architectures, complex loss functions, and advanced frameworks like diffusion models, randomly dropping tokens in a nested manner produces ordered representations that benefit AR generation without significant loss in expressivity. As shown in Table 7, reconstruction quality is preserved, indicating no degradation in representational capacity. Additionally, generative performance at 25% and 50% dropout exceeds the 0% baseline, suggesting that structured ordering facilitates AR generation, with similar results at 25% and 50% indicating that a moderate degree of nested order is sufficient.

Dropout Ratio Table 7 illustrates the influence of the channel dropout ratio on both reconstruction and generation performance. We observe that reconstruction metrics remain stable across the tested range, with no significant degradation in rFID, SSIM, or PSNR as the dropout ratio increases from 0% to 50%. However, the generation capability is substantially enhanced by the introduction of dropout; specifically, the GenEval and DPG scores show a marked improvement when moving from 0% to 25%, increasing from 0.62 to 0.74 and 72.76 to 82.14, respectively. Further increasing the dropout ratio to 50% results in comparable generation performance to the 25% setting, suggesting that a moderate dropout ratio is sufficient to optimize the model's generative robustness without sacrificing its reconstruction fidelity.

C Progressive Channel Analysis

Finally, we analyze how reconstruction changes as more channels are revealed to the decoder. We progressively keep the first n channels and mask the remaining channels before decoding. Table 8 shows a clear coarse-to-fine trend. From 32 to 128 channels, rFID, SSIM, and PSNR all improve rapidly, indicating that early channels capture global appearance, object identity, and coarse structure. After 128 channels, rFID improvement slows down, while SSIM and PSNR continue to increase steadily. This indicates that later channels mainly refine local structure and high-frequency

Table 8: **Reconstruction with progressively added channels.** We keep only the first n channels before decoding. Early channels rapidly recover global appearance and semantic structure, while later channels mainly refine local structure and texture.

# Channels	rFID ↓	SSIM↑	PSNR↑
32	30.24	0.410	16.96
64	11.84	0.472	18.38
96	6.91	0.506	19.21
128	4.05	0.524	19.82
160	3.46	0.533	20.17
192	3.03	0.539	20.39
224	2.88	0.542	20.55
256	**2.63**	**0.568**	**21.01**

texture, providing quantitative evidence that channel-wise decomposition naturally organizes visual information from global semantics to fine details.

D Variable-Resolution Extension

In the main experiments, we keep a fixed 256×256 resolution setting to enable strictly controlled comparison with VQ, yielding codewords with a spatial size of 16×16.

Here, we show a simple practice that extends CVQ to variable resolutions with lightweight resampling modules before and after quantization: before lookup, a fixed set of learnable queries cross-attend features of arbitrary spatial size ($h \times w$) to a fixed size ($h_0 \times w_0$); after lookup, we dynamically generate $h \times w$ target queries to project the quantized channels back to the decoder's desired spatial size. This plays a role analogous to the pre- and post-quantization projections commonly used in patch-wise VQ tokenizers, while preserving CVQ's channel-token formulation.

For variable-resolution training, we progressively scale input resolution from 256 to 512 and 1024. For patch-wise VQ, the embedding dim is fixed at 256, yielding token counts of 256, 1024, and 4096, respectively. For CVQ, we also fix the embedding dim to 256 (16×16) and control the token budget to the intermediate value of the VQ setting (*i.e.*, 1024) for a fair comparison.

Table 9: **Variable-resolution reconstruction.** CVQ is trained with a fixed codeword size and uses resolution-dependent resampling before and after quantization.

Method	Train Res.	Inference Res.	rFID ↓	PSNR↑
VQ	256	256	4.84	19.93
CVQ	256	256	**0.88**	**25.02**
VQ	variable	256	4.97	19.56
CVQ	variable	256	**0.92**	**24.96**
VQ	variable	512	2.05	22.01
CVQ	variable	512	**0.96**	**24.55**

Table 10: **Resolution scaling under matched token budgets.** When CVQ is allocated the same number of tokens as VQ at each resolution, its reconstruction quality also improves with resolution.

Method	Token Budget	Train Res.	Inference Res.	rFID ↓/ PSNR↑
VQ	256	variable	256	4.97 / 19.56
CVQ	256	variable	256	**2.63 / 21.06**
VQ	1024	variable	512	2.05 / 22.01
CVQ	1024	variable	512	**0.96 / 24.55**

As shown in Table 9, CVQ generalizes across resolutions and consistently outperforms the corresponding VQ baseline. When the token budget is held fixed, CVQ may slightly trade reconstruction fidelity for efficiency at higher resolutions because each token must cover a larger spatial support. This is a

controlled quality-efficiency trade-off rather than an inherent inability to model high-resolution images. As shown in Table 10, when CVQ is allocated the same token budget as VQ at each resolution, reconstruction quality improves with resolution and remains stronger than VQ.

Discussion CVQ provides a practical advantage for autoregressive generation by decoupling token count from image resolution. While patch-wise methods exhibit quadratic token growth as the latent grid scales, CAR maintains nearly constant inference cost by keeping the channel sequence length fixed. Instead of increasing token count, CVQ encodes spatial resolution within each channel token.

E Training Setup and Data Sources

The CVQ tokenizer follows the VQGAN-style encoder-decoder architecture [10] and is trained on ImageNet-1K at 256×256 resolution. With a downsampling factor of $f = 16$, CVQ quantizes each channel of a $16 \times 16 \times c$ latent feature map using a shared codebook, where $c \in \{256, 1024\}$. Detailed configurations are provided in Table 11.

Table 11: **CVQ tokenizer configuration and training recipe.**

Setting	Value
Dataset	ImageNet-1K
Input resolution	256×256
CVQ latent shape	$16 \times 16 \times 256$ or $16 \times 16 \times 1024$
Codeword dimension	256
Codebook size	16,384
Discriminator	PatchGAN
Optimizer	Adam ($\beta_1 = 0.5$, $\beta_2 = 0.9$)
Learning rate / weight decay	10^{-4} / 10^{-4}
Batch size / epochs	256 / 100
Training hardware	8 NVIDIA H20 GPUs

For CAR tokenizers, nested channel dropout is applied with probability $\alpha = 0.25$ to induce coarse-to-fine ordering. The retained prefix length is uniformly sampled from $\{1, \ldots, c\}$, with remaining channels zeroed before decoding and quantization loss computed only over active channels. Further details on the training objective are provided in Appendix B.

CAR is trained in two stages using AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.96$, weight decay 10^{-3}). Stage I trains only the MLP projector and LLM head at a learning rate of 10^{-4}, keeping the LLM backbone frozen. Stage II jointly optimizes all components at a learning rate of 2×10^{-5}. The Training data comprises a filtered ImageNet-21K subset, LAION-Aesthetics-12M, CC12M [5], Megalith-10M[4], BLIP-3o-short/long [6], and 6M in-house aesthetic images.

[4]Re-captioned by Qwen3-VL.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, we have discussed the current limitations of the work in future works.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, we disclose the information needed to reproduce the main experimental results of the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: See supplemental materials.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, we specify all the training and test details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to the large computational cost of training image generation models and their tokenizers, we do not report error bars or repeated-run statistics. We evaluate multiple settings in Sec. 4, making repeated runs for every setting prohibitively expensive. This is in line with prior work in the image generation domain.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Yes, the authors have reviewed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See Appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We will control the use rights of the model after it is released.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cite all the original paper of assets we used.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: See supplementary materials.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as an important, original components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.