CODA: Repurposing Continuous VAEs for Discrete Tokenization

Zeyu Liu[1*] Zanlin Ni[1*] Yeguo Hua[1] Xin Deng[2]
Xiao Ma[3] Cheng Zhong[3] Gao Huang[1†]
[1] Tsinghua University [2] Renmin University [3] Lenovo Research, AI Lab
https://lzy-tony.github.io/coda

Abstract

Discrete visual tokenizers transform images into a sequence of tokens, enabling token-based visual generation akin to language models. However, this process is inherently challenging, as it requires both compressing *visual signals into a compact representation and* discretizing *them into a fixed set of codes. Traditional discrete tokenizers typically learn the two tasks jointly, often leading to unstable training, low codebook utilization, and limited reconstruction quality. In this paper, we introduce* ***CODA*** *(**CO**ntinuous-to-**D**iscrete **A**daptation), a framework that decouples compression and discretization. Instead of training discrete tokenizers from scratch, CODA adapts off-the-shelf continuous VAEs—already optimized for perceptual compression—into discrete tokenizers via a carefully designed discretization process. By primarily focusing on discretization, CODA ensures stable and efficient training while retaining the strong visual fidelity of continuous VAEs. Empirically, with* $\mathbf{6}\times$ *less training budget than standard VQGAN, our approach achieves a remarkable codebook utilization of **100%** and notable reconstruction FID (rFID) of* $\mathbf{0.43}$ *and* $\mathbf{1.34}$ *for* $8\times$ *and* $16\times$ *compression on ImageNet 256*$\times$ *256 benchmark.*

1. Introduction

The field of AI-generated content (AIGC) has witnessed significant progress in recent years. In natural language processing, text generation has been mainly unified by the paradigm of next discrete token prediction [1, 6, 35, 45]. This contrasts with developments in computer vision, where the debate between *continuous* and *discrete* generation paradigms has yet to reach a conclusion. While continuous diffusion models dominate the field [5, 31, 32, 37], discrete token-based approaches [7, 8, 29, 41, 42] are increasingly gaining interest due to their computational efficiency [28, 30, 42] and promising potential for unifying tasks across language modeling and multimodal understanding [47, 50, 52, 53].

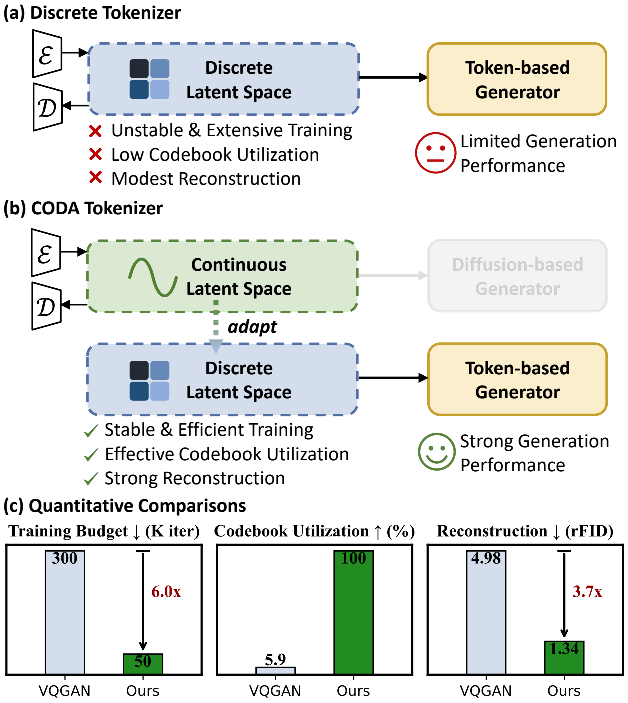


Figure 1. (a) **Conventional discrete VQ tokenizers** learn to *compress* and *discretize* inherently continuous visual signals into codes simultaneously. This lead to multiple challenges in training and the corresponding unsatisfactory latent space poses a bottleneck that limit the performance of discrete token-based generation models. (b) **Our proposed CODA tokenizer** leverages continuous VAEs for compression, directly discretizing the latent space. (c) **Quantitative comparisons** between VQGAN [11] and our proposed CODA tokenizer.

One defining characteristic of discrete methods is the need for a discrete tokenizer, which converts visual signals into a discrete format akin to language tokens. This process, however, is non-trivial, as it requires simultaneously *compressing* visual signals and *discretizing* them into a set of codes. In contrast, continuous VAEs focus solely on mapping visual signals into a compressed continuous latent space, avoiding this added complexity. Previous research

[*] Equal contribution.
[†] Corresponding Author.

has identified various problems in standard VQ-based discrete tokenizers, including unstable training [3, 54, 61], low codebook utilization [12, 16, 60], and limited performance even under extensive training [11, 49]. For example, while VAE used in state-of-the-art diffusion models [5] achieves 0.17 reconstruction FID (rFID) on ImageNet [9], the standard VQGAN [11] tokenizer achieves only 4.98 rFID. In response to these challenges, recent research has explored many advanced techniques for discrete tokenization, such as embedding-free tokenization [25, 26, 49, 55], rotation trick [12], and 1D tokenization [13, 57].

Despite these recent advances, discrete tokenizers are still required to learn compression and discretization concurrently and are often regarded as mutually exclusive from continuous VAEs. In this paper, we challenge this dichotomy and propose a novel **CODA** (**CO**ntinuous-to-**D**iscrete **A**daptation) framework that decouples compression and discretization. Instead of training a discrete tokenizer to handle compression and discretization in tandem, we demonstrate that off-the-shelf continuous VAEs, which are already highly optimized for perceptual compression, can be directly adapted into discrete tokenizers through a carefully designed discretization process (see Tab. 1 and Figure 2). In this way, CODA retains the strong visual fidelity of continuous VAEs while enabling discrete representation through a secondary transformation. Moreover, by focusing primarily on discretization, our method ensures more stable and efficient training, leading to significantly higher codebook utilization and improved reconstruction performance.

Building upon this conceptual framework, we craft a set of CODA tokenizers, instantiated through systematic adaptations on the $16\times$ compression MAR [21] and $8\times$ compression FLUX [5] VAEs. Quantitative evaluations demonstrate that CODA achieves competitive performance of 1.34 and 0.43 rFID on the ImageNet dataset, respectively, while maintaining full codebook utilization. Moreover, our proposed disentangling leads to high computational efficiency in training, enabling a reduction of $6\times$ in training compute compared to representative prior VQGAN training settings. Empirical results also demonstrate that, when integrated with the CODA tokenizer, representative discrete token-based generation methods, *i.e*. MaskGIT [7] can be significantly improved to achieve overall competitive performance against main stream continuous and discrete generation paradigms.

Our main contributions can be summarized as follows:

1. We challenge the conventional dichotomy of continuous VAEs and discrete tokenizers, and present a novel prospective to designing discrete tokenizers by leveraging continuous VAEs for compression while learning solely effective discretization.
2. Motivated by this approach, we craft a series of CODA tokenizers, reaching competitive reconstruction quality while requiring only a fraction of training time compared to standard settings.
3. Combining with MaskGIT, we demonstrate that our CODA tokenizer unlocks enhanced and competitive performance in discrete token-based image synthesis.

2. Related Work

2.1. Continuous and Discrete Image Synthesis

Continuous approaches to image synthesis, i.e. latent diffusion models [5, 31, 32, 37] operate on a latent space sampled from a Variational Autoencoder (VAE) [18]. Images are first mapped from pixel space into this compressed representation and then subjected to iterative denoising via diffusion models. At each step, a trained neural network refines the latent by removing noise and rebuilding semantic structure and fine details. Once denoising is complete, the latents are decoded back into pixel space to generate the final image. This approach has been proven highly successful, enabling the generation of high-quality, photorealistic images.

Discrete approaches to image synthesis utilize discrete token representations as means for generative modeling. Images in pixel space are first transformed using image tokenizers [11, 44] into discrete tokens. Multiple paradigms build upon this discrete space and iteratively generate the predicted token combination, after which the tokens are decoded back to pixel space. Autoregressive [19, 41, 56] models draw inspiration from language modeling, treating images as a sequence of discrete tokens and generating them by iteratively predicting the next token in the sequence. MaskGITs [7, 8, 29] models leverage the bi-directional nature of images by enabling parallel decoding. These models simultaneously unmask a combination of discrete tokens at each forward step, improving the efficiency and speed of the generation process. Visual autoregressive [42] models are inspired by the coarse-to-fine approach in image synthesis and adopt next-scale prediction to predict a combination of discrete tokens from the same scale at each step. Discrete token-based generation is gradually gaining popularity and becoming a promising approach to visual synthesis, as it is more efficient in terms of inference speed, aligns and adapts to similar paradigms in natural language processing [33, 34, 43], multimodal understanding [22, 47, 53] and Embodied AI [51].

2.2. Discrete Image Tokenizers

Image tokenizers play a foundational role in discrete visual generation. Extensive works have been dedicated to improving performance, training stability, compressive representation and scalability. The foundational work of VQVAE [44] and VQVAE2 [36] introduced the paradigm of neural discrete representation learning, employing straight-through estimators to train quantized models in an end-to-end framework. Building upon this, VQGAN [11] refines the training recipe of VQVAE, improving the fidelity of detailed image reconstruction and enabling high-resolution image generation. RQ-VAE [19] further proposes residual quantization to reduce quantization error caused by discrete tokenization. FSQ [26] and LFQ [25, 55] investigate the impact of embedding-free tokenization and scaling codebook size to at most 2^{18}. VQGAN-LC [60] addresses the challenges in scaling codebook sizes by incorporating both pretrained vision features as codebook entries and training projectors to mitigate low utilization rates and mode collapse. IBQ [39] introduces joint optimization of all codebook entries during each forward-backward pass, further enhancing performance with large codebook sizes. In a parallel research trajectory, SEED [13] and TiTok [57] explore 1D tokenization, which facilitate high compression ratios.

3. Preliminaries of Discrete Vector Quantization

In this section, we give a preliminary overview of discrete vector quantization. Vector quantization is a classical method which aims to project and compress visual signals into a discrete latent token space formulated by a fixed codebook $\mathcal{C} \in \mathbb{R}^{n\times d}$, where n and d are the size and dimension, respectively. A typical vector quantized tokenizer consists of three modules: an encoder $\mathcal{E}$, decoder $\mathcal{D}$ and quantizer $\mathcal{Q}$. For a given image $\mathcal{I} \in \mathbb{R}^{3\times H\times W}$, it is first encoded by the encoder $\mathcal{E}$ into a compressed feature map $f \in \mathbb{R}^{h\times w\times d}$, then projected by $\mathcal{Q}$ into z in the discrete latent space spanned by $\mathcal{C}$. Finally, z is decoded back to pixel space using $\mathcal{D}$.

As the key component for discretization, the vector quantizer $\mathcal{Q}$ typically assigns features to codes by selecting the nearest neighbor z based on Euclidean distance in $\mathbb{R}^d$ space.

$$z = C_k \in \mathbb{R}^d, k = \arg\min_{c\in C} \|f - c\| \tag{1}$$

During training, the non-differentiable $\arg\min$ operator poses a challenge in estimating gradients. To circumvent this, the straight-through estimator (STE) [4] is employed to backpropagate the gradients back to the encoder, ensuring that the encoder and selected code are updated simultaneously.

$$z_{\text{quant}} = f + \text{sg}[z - f] \tag{2}$$

where $\text{sg}[]$ is the stop-gradient operation.

This basic quantization approach has several limitations: Simultaneously updating the encoder, decoder, and quantizer leads to unstable training and makes the model highly sensitive to hyper-parameters and restart schemes, as the encoder, decoder and codebook must adapt to each other's evolving distributions [3, 54, 61]. Assigning codes based solely on Euclidean distance can cause codebook collapse and inefficient code utilization as the codebook size increases [12, 16, 60]. Furthermore, the training process requires considerable time and is slow to converge [49].

Method	Quant. Err$\downarrow$	rFID$\downarrow$	PSNR$\uparrow$	SSIM$\uparrow$	Util.$\uparrow$
Continuous VAE [21]	-	0.69	24.9	0.716	-
Vector Quant.	0.181	41.52	18.9	0.461	49%
+ Residual Quant.	0.094	14.23	20.6	0.533	51%
+ Attention-based Quant.	0.065	7.06	21.4	0.571	100%
+ Adapt VAE	0.001	1.34	22.2	0.602	100%

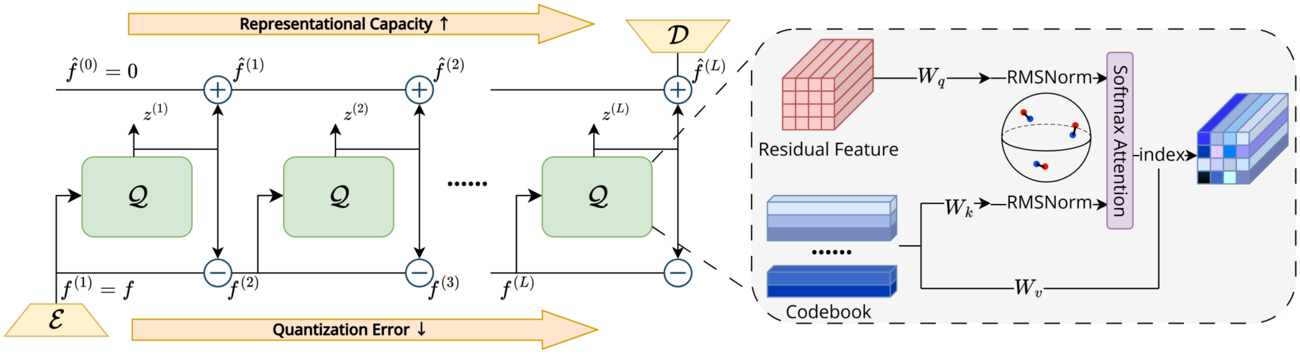
Table 1. **Summary of our main ablation results in building the CODA tokenizer.** We start from an off-the-shelf continuous VAE [21] (marked in gray) and progressively introduce design components for effective tokenization. The codebook size is fixed to 65536. Quant: quantization. Util: utilization of codebook.

4. CODA Tokenizer

Given a continuous VAE, a straightforward solution for converting it into a discrete tokenizer is to directly employ well-established quantization methods, *e.g.* vector quantization (VQ) to its latent space. Specifically, as discussed in Section 3, a set of codebook embeddings can be initialized and optimized to approximate the continuous latents at their best. However, this naive approximation incurs significant performance drop compared to the original continuous VAE, as shown in Tab. 1. In this section, we carefully inspect the difficulties in discretizing a continuous VAE, and present corresponding solutions step by step.

Problem I: insufficient representational capacity. To analyze the reason behind the performance drop, we first visualize the continuous latent space of the pretrained VAE and the discretized space after applying vector quantization in Figure 3 (a) and (b). Results indicate a substantial information loss during the VQ approximation process. More specifically, continuous features span across large domains in the latent space, covering not only high-density areas in the center but also marginal regions that are less populated. In contrast, discrete features quantized by VQ occupy only sparse points in the latent space. This outcome is not surprising, though, as the number of features that VQ can represent is inherently limited by the number of code embeddings in codebook. Though we have already employed an extensive codebook size of 65536, exceeding that of many representative VQ tokenizers [11, 41, 49, 54], the representational capacity still remains far from sufficient to capture the full diversity of the densely populated continuous latent space.

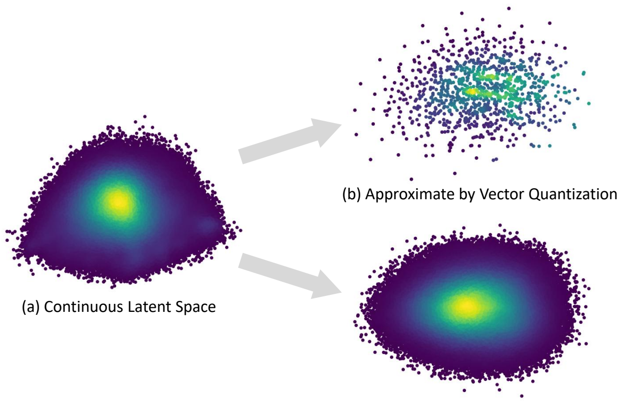
Solution: residual quantization. To overcome the representational limitations inherent in vector quantization, we draw inspiration from similar tasks that approximate continuous functions in numerical analysis. A common strategy to enhance the representational capacity of estimations involves decomposing the target function into the combination of multiple basis functions. For instance, k-th spline interpolation decomposes a continuous function $f(x)$ and represent it using $k + 1$ discrete coefficients, where increasing the level k corresponds to enhanced representation capacity, allowing



(a) Residual Quantization

(b) Attention-based Quantization

Figure 2. **Illustration of our CODA tokenizer.** (a) a residual quantization process of L levels iteratively refines the approximation of a continuous VAE vector f through a composite of multiple quantization layers, thus progressively minimizing the quantization error. Meanwhile, as the continuous VAE vector is approximated by a *combination* of L discrete codes, the representational capacity is significantly enlarged. (b) the attention-based quantization process frames discretization as a *retrieval* task. Continuous features and codebook embeddings are projected and normalized onto a normed hypersphere, where the softmax attention matrix is computed to determine the confidence of code selection. As codes compete within the softmax attention framework, this approach ensures a *sparse* and *unambiguous* assignment.



(a) Continuous Latent Space

(b) Approximate by Vector Quantization

(c) Approximate by Residual Quantization

Figure 3. **Visualization of latent space approximation**: (a) the original latent space of the continuous VAE, (b) latent space approximated by vector quantization and (c) latent space approximated by residual quantization.

for more precise approximations of the original function:

$$f_i(x) \approx \sum_{l=0}^{k} z_{i,l}(x - t_i)^l, x \in [t_i, t_{i+1}] \tag{3}$$

Inspired by this, we employ a conceptually similar technique, namely residual quantization [19] in place of vector quantization. Residual quantization progressively refines the approximation of a continuous VAE vector f through a composite of multiple quantization layers, with each layer building upon the residual of the previous one:

$$\varepsilon_{l+1} = \varepsilon_l - z^{(l)} \tag{4}$$

where ε_l and $z^{(l)}$ represents the approximation error and discrete code values at the l-th layer. In this way, quantization error can be iteratively minimized. At the same time, since the continuous vector is now represented by a *combination* of discrete tokens at L levels:

$$f \approx \sum_{l=1}^{L} z^{(l)} \tag{5}$$

the number of possible combinations grows exponentially with more quantization levels, thus providing a significantly larger representational capacity.

We explore the effect of residual quantization on approximating a continuous latent manifold. As shown in Figure 3, the latent space generated by residual quantization closely resembles that of continuous latent spaces, suggesting an increased representational capacity. As shown in Figure 4, the quantization process progressively refines at each level, leading to a consistent reduction in quantization error, which corresponds to a similar decrease in rFID. Finally, when the number of quantization levels reaches 10, the quantiza-

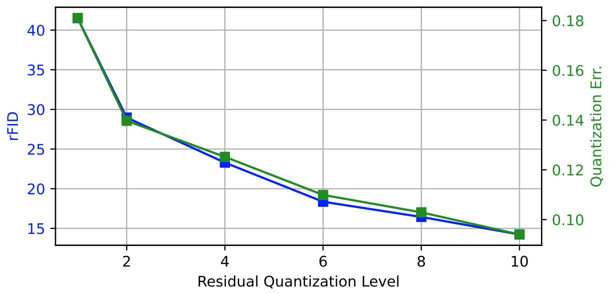
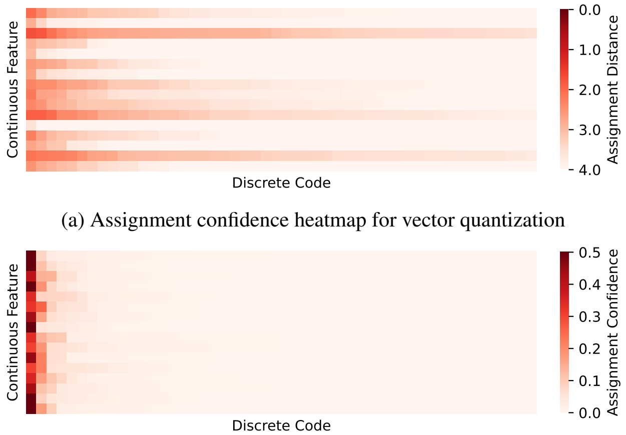


Figure 4. **Effect of residual quantization levels on tokenizer performance.** With more levels of residual quantization, quantization error is consistently minimized, and the reconstruction performance (measured by rFID) steadily improves.



(a) Assignment confidence heatmap for vector quantization

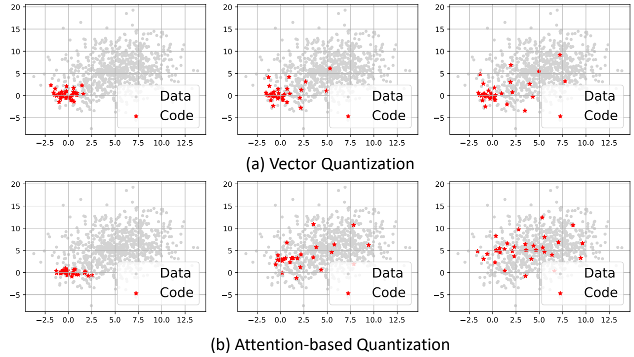
(b) Assignment confidence heatmap for attention quantization

Figure 5. **Visualization of top assignment confidence scores** for 16 randomly selected continuous VAE features. For vector quantization, we visualize the distance of codes to the continuous feature, with lower distance representing higher confidence.

tion error drops from 0.181 to 0.094, while rFID improves significantly from 41.52 to 14.23, as reported in Tab. 1.

Problem II: ambiguous code assignment. Although residual quantization has significantly improved the precision of latent space approximation, thereby enhancing reconstruction performance, we find that codebook utilization remains insufficient. As shown in Tab. 1, nearly half of the codes remain unused during quantization. To further investigate this, we randomly select 16 continuous VAE features and for each, we visualize the Euclidean distance of its nearest 50 neighbors from left to right. Results reveal a clear pattern of *ambiguity* in code assignment. Specifically, for some features, the assignment choice has minimal impact, as these features are similarly related to multiple code vectors, leading to effective approximation regardless of assignment. Conversely, other features are far from all codes, resulting in high quantization error irrespective of assignment choice. Consequently, the imbalance in code distribution contributes to insufficient utilization of the codebook.

Solution: attention-based quantization. We hypothesize that this ambiguity arises from the lack of ***sparsity*** in the quantization process: Ideally, *sparse* assignment schemes ensure that a continuous feature is strongly associated with a single discrete code with high confidence, while remaining weakly associated to all other codes with low confidence. However, in VQ based approaches, assignments are not sparse, as features are often closely related to multiple codes in Euclidean space. This lack of sparsity makes the assignment highly sensitive to small variations in distance, resulting in ambiguous assignment. During training, the selected codebook indices are simply *pulled* toward their respective clustering centroids. However, due to the lack of sparsity, multiple codes may be drawn toward the same centroid, resulting in under-utilization of the codebook. As a result, codes tend to converge around limited centers, rather than adequately covering the entire latent space. Visualization of vector quantization training dynamics, shown in Figure 6 (a), confirms this process, where only a few codes are spread across the latent space while others form a single clique.



(a) Vector Quantization

(b) Attention-based Quantization

Figure 6. **Visualization of training dynamics**. In attention quantization, codes are *pushed* to fully occupy the latent space, whereas vector quantization shows limited coverage of latent space.

Motivated by these observations, we propose a novel quantization mechanism that promotes sparse assignment, and further reduces quantization errors at each quantization level. Drawing inspiration from the enhanced sparsity observed in softmax attention schemes, we design a learnable attention-based assignment strategy, as illustrated in Figure 2 (b). Given an encoded feature $\mathbf{F}$ and codebook $\mathbf{C}$, the quantizer maps them to a distance space using weights $\mathbf{W}_q$ and $\mathbf{W}_k$, respectively. The projected weights are then normalized using RMSNorm [58], which projects the features and clusters onto a normed hypersphere [23], and multiplied to compute the distance matrix $\mathbf{A}$ using softmax attention.

$$\mathbf{Q} = \text{rms\_norm}(\mathbf{F}\mathbf{W}_q) \quad \mathbf{K} = \text{rms\_norm}(\mathbf{C}\mathbf{W}_k) \tag{6}$$

$$\mathbf{A} = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}) \tag{7}$$

where d is the hidden dimension. Feature at each position is assigned to the index with largest attention (*i.e.* similarity or distance) in normed space. The assignment is propagated to the codebook weights mapped by projection $\mathbf{W}_v$:

$$\hat{\mathrm{F}} = \text{one\_hot}(\mathbf{A})^T(\mathbf{C}\mathbf{W}_v) \tag{8}$$

This design introduces a learnable mechanism for assigning features to discrete codes with enhanced sparsity and improved training dynamics. Attention-based quantization frames discretization as a *retrieval task*, where the quantizer learns to retrieve the most appropriate index from a large codebook. Under this framework, different codes compete and suppress each other when calculating softmax attention scores, enforcing sparse assignment. This is confirmed by the visualization of confidence in Figure 5b (b). Additionally, this approach delivers a *push* effect along with *pull* during training: the chosen code index is *pulled* towards the clustering centroids, reducing quantization error, while irrelevant codes are *pushed* away to other regions of the latent space. This new dynamic ensures more effective codebook usage and comprehensive coverage of the latent space, as visualization of training dynamics in Figure 6 (b) illustrate. Quantitative results in Tab. 1 shows that our the attention-based assignment achieves a remarkable *100%* codebook utilization, and significantly improves the reconstruction FID from 14.23 to 7.06.

Adapting VAE to the learned discrete space. Beyond approximating the continuous latent space with discrete codes, we find that slightly adjusting the original VAE parameters to accommodate the distribution shift induced by discretization is also beneficial. Specifically, we incorporate LoRA [14] parameters into the continuous VAE, allowing it to evolve in tandem with the learning of the discretized latent space. As shown in Tab. 4a, this adaptation reduces quantization error to 0.001 and enhances the reconstruction quality, measured by rFID, to 1.34. This significantly bridges the gap with the original continuous VAE, which achieves an rFID of 0.69.

Implementation details. We follow the standard VQGAN [11] design and train our tokenizer with a mixed combination of losses:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_p \mathcal{L}_p + \lambda_q \mathcal{L}_q + \lambda_{\text{adv}} \mathcal{L}_{\text{adv}} + \lambda_e \mathcal{L}_e \tag{9}$$

where $\mathcal{L}_{\text{rec}}$ is the pixel-wise reconstruction loss, $\mathcal{L}_p$ is the LPIPS [59] perceptual loss, $\mathcal{L}_{\text{adv}}$ is an adversarial GAN loss passed through a StyleGAN-T discriminator [38], $\mathcal{L}_e$ is an regularization entropy penalty for encouraging codebook usage. $\mathcal{L}_q$ is a combination of soft and hard commitment and quantization loss proposed by [39]:

$$\begin{aligned} \mathcal{L}_q =& \|\text{sg}[z_{\text{hard}}] - f\|^2 + \beta \|z_{\text{hard}} - \text{sg}[f]\|^2 & (10) \\ &+ \|z_{\text{soft}} - f\|^2 & (11) \end{aligned}$$

We calculate the quantization loss $\mathcal{L}_q$ and entropy loss $\mathcal{L}_e$ at each level, encouraging codes at each level to minimize the quantization error and boosting codebook utilization.

Token-based generation. To verify the effectiveness of our improved tokenizer, we integrate it with the representative token-based visual generation framework MaskGIT [7]. MaskGIT operates on a sequence of tokens $\mathbf{z}$ compressed by a quantized autoencoder. During training, MaskGIT randomly mask out a set of tokens using the special mask token [MASK]. Based on the unmasked set $\bar{\mathbf{M}}$, the model predicts the logits for masked token set $\mathbf{M}$, then optimizes a BERT-style [17] Masked Language Modeling (MLM) Loss

$$\mathcal{L}_{\text{MLM}} = - \sum_{i \in [1,N], m_i = 1} \log p(z_i | z_{\bar{\mathbf{M}}}) \tag{12}$$

where $p(z_i | z_{\bar{\mathbf{M}}})$ is the predicted logits of token index z_i based on the unmasked token set $\bar{M}$.

During inference, MaskGIT start from a fully masked sequence and decodes iteratively. During each step, the model predicts logits for all masked positions and samples a portion of most confident tokens to be unmasked according to a predefined schedule. The final fully unmasked sequence is then decoded back to pixel space using the tokenizer decoder.

5. Experiments

5.1. Experiment Setup

Datasets and evaluation metrics. We conduct our experiments on the ImageNet dataset [9], specifically utilizing images at a resolution of 256 × 256. Our study involves both training our tokenizer and evaluating generative models using this dataset. To assess the reconstruction quality of our approach, we employ multiple evaluation metrics, including reconstruction FID (rFID), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index Measure (SSIM). These metrics provide a comprehensive evaluation of both perceptual and pixel-level reconstruction accuracy.

Tokenizer training. Leveraging the continuous latent space of pretrained VAEs, we utilize and freeze the encoder-decoder weights from two standard continuous VAE: a 16× MAR [21] VAE and a 8× FLUX VAE [5]. To adapt the continuous encoder-decoder to discrete codes, we implement LoRA [14] modules exclusively on the convolution blocks and finetune the added modules along with the quantizer during training. In practice, we adopt a LoRA rank of 32 for both encoder and decoder. We follow a main stream implementation of residual quantization introduced in [42] and set the quantization level to 10. All models are trained with an accumulated batch size of 512 for 10 epochs on the ImageNet 256 × 256 dataset with a $5e-4$ learning rate.

Generative model training. MaskGIT models are trained for 500K steps with an accumulated batch size of 2048 on the ImageNet 256 × 256 dataset with $4e-4$ learning rate. In addition to training on the standard tokenized representation, we introduce learnable positional embeddings specific to each quantization level. This modification enhances the model's ability to differentiate between tokens from different levels of the residual quantization hierarchy, ultimately improving the decoding process.

5.2. Main Results

Tokenizer performance. We present the reconstruction results in Tab. 2. There are three key observations from our findings: First, our CODA tokenizer achieves competitive performance with 0.43 and 1.34 rFID in 8× and 16× reduction settings, respectively, significantly outperforming corresponding baselines in terms of reconstruction quality and fidelity. Second, our tokenizer achieves a high utilization

Method	Ratio	Codebook Size	rFID ↓	PSNR ↑	SSIM ↑	Utilization (%)	Training (iter)
VQGAN [11]	16	1024	7.94	19.4	0.500	44	300K
VQGAN [11]	16	16384	4.98	19.9	0.510	5.9	300K
DF-VQGAN [27]	16	12288	5.16	-	-	-	-
DQVAE [15]	16	1024	4.08	-	-	-	250K
DiVAE [40]	16	16384	4.07	-	-	-	-
VQGAN-LC [60]	16	16384	3.01	23.2	0.564	99	100K
VQGAN-LC [60]	16	100000	2.62	23.8	0.589	99	100K
LlamaGen [41]	16	16384	2.19	20.8	0.675	97	200K
RQVAE [19]	-	16384	2.69	-	-	-	250K
BAE [48]	16	65536	3.32	-	-	-	2000K
MaskBit [49]	16	16384	1.61	-	-	-	1350K
CODA MAR	16	16384	1.43	22.0	0.594	100	**50K**
CODA MAR	16	65536	**1.34**	22.2	0.602	100	**50K**
MAR [21] VAE	16	-	0.69	24.9	0.716	-	-
VQGAN† [11]	8	16384	1.14	23.4	0.670	5.4	-
ViT-VQGAN [54]	8	8192	1.28	-	-	-	-
DF-VQGAN [27]	8	8192	1.38	-	-	-	-
DiVAE [40]	8	16384	1.28	-	-	-	-
OmniTokenizer† [46]	8	8192	1.11	24.0	0.752	100	-
CODA FLUX	8	65536	**0.43**	**25.9**	**0.771**	100	**50K**
FLUX [5] VAE	8	-	0.17	31.1	0.903	-	-

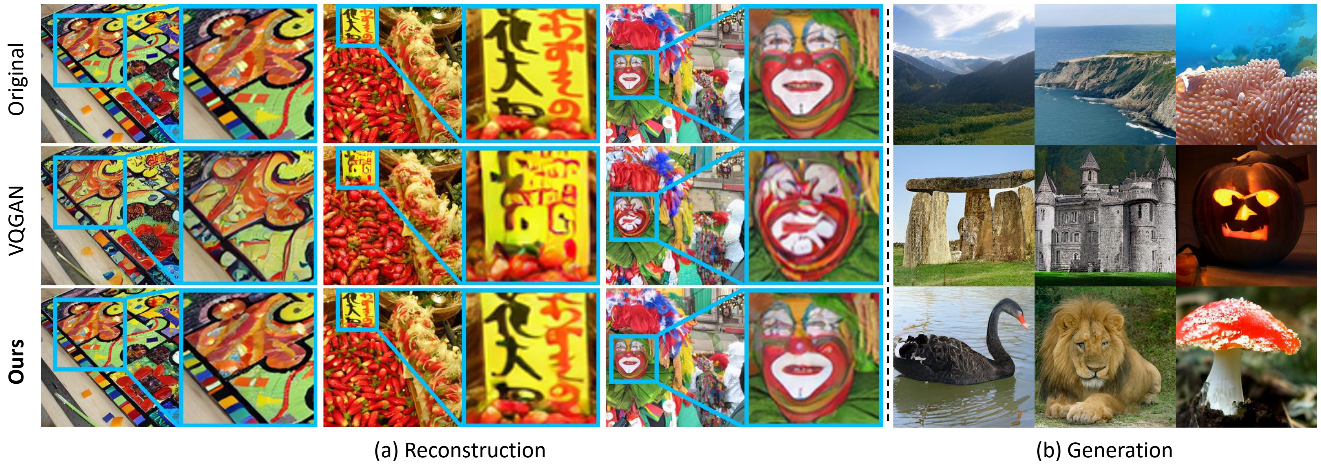
Table 2. **Quantitative reconstruction results on ImageNet 256×256.** Training iterations are uniformly converted and measured under global batch size of 256 following [49]. Ratio represents compression rate between image and latent resolution. Results from referenced continuous VAEs are marked in gray. † denotes training enhanced by additional web-scale data other than ImageNet.

rate of 100% with a large codebook of 65536, compared to VQGAN's utilization rate of 5.9% with 16384 codes. This demonstrates that the combined approach of training in a pretrained continuous latent space and employing attention-based sparse quantization mechanisms prevents codebook collapse and encourages high utilization, providing the potential for scaling to even larger scopes. Finally, our approach requires substantially less training computation than all baseline methods, achieving a 6× speedup compared to the representative VQGAN training recipe. Additional qualitative visualizations and comparisons are provided in Figure 7 (a).

Image generation. In Tab. 3, We compare our approach with leading image generation paradigms, including continuous diffusion-based methods, discrete MaskGIT, autoregressive models, and visual autoregressive models. Our key observations are as follows: 1) Compared to other MaskGIT-based models, our approach achieves a lower FID of 3.17, compared to 4.92 for MaskGIT and 4.53 for MaskGIT-FSQ, demonstrating enhanced generation quality delivered by a more effective tokenization strategy. 2) Our model demonstrates strong overall performance relative to other main stream generation paradigms, with an FID of 2.66 compared to 3.60 for LDM and 3.07 for LlamaGen. 3) Our approach achieves high-quality generation results comparable to continuous methods while maintaining the superior efficiency of discrete models, requiring fewer sampling steps. Additional quantitative text-to-image generation results and qualitative results of generated images from our approach are demonstrated in Tab. 5 and Figure 7 (b) respectively.

Method	Type	#Params	Steps	FID↓
ADM [10]	Diff.	554M	250	10.94
LDM-4-G [37]	Diff.	400M	250	3.60
DiT-L/2 [31]	Diff.	458M	250	5.02
UViT-L/2 [2]	Diff.	287M	50	3.40
LDM-4-G† [37]	Diff.	400M	8	4.56
DiT-XL/2† [31]	Diff.	675M	8	5.18
RQVAE-GPT [19]	AR	480M	256	15.7
VQGAN-LC-GPT [60]	AR	404M	256	15.4
ViT-VQGAN-GPT [54]	AR	650M	1024	8.81
RQ-Transformer [19]	AR	3.8B	64	7.55
LlamaGen-L-384 [41]	AR	343M	576	3.07
VAR [42]	VAR	310M	10	3.30
MaskGIT [7]	Mask.	227M	12	4.92
MaskGIT-FSQ [26]	Mask.	225M	12	4.53
MAGE [20]	Mask.	230M	20	6.93
ENAT [28]	Mask.	219M	8	3.53
CODA MaskGIT-L	Mask.	195M	8	3.17
CODA MaskGIT-L	Mask.	195M	32	**2.66**

Table 3. **Quantitative generation results on ImageNet 256×256.** ”-384” denotes images generated at 384 resolution and resized back to 256 during evaluation. † denotes diffusion schedule augmented by DPM-Solver [24].



(a) Reconstruction

(b) Generation

Figure 7. **Visualization of samples on ImageNet** 256 × 256. (a) Reconstruction results by CODA tokenizer. Compared with VQGAN, CODA showcases higher fidelity and effectively preserves rich details. (b) Generated samples by combining CODA with MaskGIT.

Enc.	Dec.	rFID↓	PSNR↑	SSIM↑
✗	✗	7.06	21.4	0.571
✓	✗	5.75	21.3	0.551
✗	✓	1.59	22.1	0.604
✓	✓	1.34	22.2	0.602

(a) Comparison on effect of tuned weights.

Codebook Size	rFID↓	PSNR↑	SSIM↑
1024	2.07	21.2	0.561
4096	1.76	21.7	0.582
16384	1.43	22.0	0.594
65536	1.34	22.2	0.602

(b) Comparison on effect of codebook size.

Norm	rFID↓	PSNR↑	SSIM↑
w/o	1.87	21.51	0.570
LayerNorm	1.37	22.1	0.600
RMSNorm	1.34	22.2	0.602

(c) Comparison on effect of normalization.

Table 4. **Ablation studies.** We use CODA-MAR tokenizer and mark our default setting in gray

Method	VQ-Diffusion	U-Net	U-ViT-S	CODA MaskGIT-S
FID↓	13.86	7.32	5.95	**5.44**

Table 5. Text-to-image generation on MS-COCO.

5.3. Ablation Studies

In this section, we present more ablation studies to justify the effectiveness of our approach.

VAE parameters. We investigate the impact of fine-tuning the encoder and decoder LoRA weights when adapting a continuous VAE into a discrete tokenizer. To this end, we implement tokenizer training with different configurations: freezing both the encoder and decoder, training LoRA modules exclusively on the encoder, training LoRA modules exclusively on the decoder, and training LoRA modules on both the encoder and decoder. The results are presented in Table 4a. Interestingly, we observe that adapting the encoder and decoder affect performance differently. Specifically, training a small number of parameters on the decoder leads to noteworthy improvements, whereas training on the encoder yields only marginal gains. This observed asymmetry in training modules align with our motivation that approximating the discrete model closely to the continuous latent space is sufficient to achieve competitive performance.

Effect of scaling codebook size. Results in Tab. 4b demonstrate that our approach enables stable scaling of the codebook size from 1024 to 65536, leading to a consistent improvement in tokenizer performance, with rFID steadily decreasing from 2.07 to 1.34. This highlights the effectiveness of our method in fully utilizing an expanded codebook to refine discrete representations without encountering common issues such as codebook collapse or under-utilization.

Effect of normalization. We argue that normalization used in the attention quantization modules play a positive role in better assignment, leading to improved reconstruction results. As results in Tab. 4c demonstrate, projecting the query-key pairs into a normed space improves rFID from 1.87 to 1.34. Other types of normalization, *e.g*. LayerNorm is also capable of achieving similar results compared to using RMSNorm.

6. Conclusion

In this work, we present CODA tokenizers, a novel approach for training discrete tokenizers by leveraging the pretrained space of continuous VAEs for compression, while optimizing discretization through principled adaptations. Through a series of carefully designed discretization mechanisms, CODA is able to approximate continuous latent space using discrete tokens. Experiments on ImageNet demonstrate that CODA enables improved high fidelity reconstruction performance with effective codebook utilization, while requiring only minimal training. Additional experiments reveal that CODA unlocks new potential for enhanced discrete generation.

Acknowledgement

This work is supported in part by the National Key R&D Program of China (2024YFB4708200), the National Natural Science Foundation of China (U24B20173, 62321005), and the Scientific Research Innovation Capability Support Project for Young Faculty (ZYGXQNJSKYCXNLZCXM-I20).

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. 1

[2] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22669–22679, 2023. 7

[3] Gulcin Baykal, Melih Kandemir, and Gozde Unal. Edvae: Mitigating codebook collapse with evidential discrete variational autoencoders. *Pattern Recognition*, 156:110792, 2024. 2, 3

[4] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. 3

[5] BlackForest. Black forest labs; frontier ai lab, 2024. 1, 2, 6, 7

[6] 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*, 33:1877–1901, 2020. 1

[7] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022. 1, 2, 6, 7

[8] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, José Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. In *Proceedings of the 40th International Conference on Machine Learning*, pages 4055–4075, 2023. 1, 2

[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*, pages 248–255. Ieee, 2009. 2, 6

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 7

[11] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 1, 2, 3, 6, 7

[12] Christopher Fifty, Ronald G Junkins, Dennis Duan, Aniketh Iger, Jerry W Liu, Ehsan Amid, Sebastian Thrun, and Christopher Ré. Restructuring vector quantization with the rotation trick. *arXiv preprint arXiv:2410.06424*, 2024. 2, 3

[13] Yuying Ge, Yixiao Ge, Ziyun Zeng, Xintao Wang, and Ying Shan. Planting a seed of vision in large language model. *arXiv preprint arXiv:2307.08041*, 2023. 2, 3

[14] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*. 6

[15] 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*, pages 22596–22605, 2023. 7

[16] Minyoung Huh, Brian Cheung, Pulkit Agrawal, and Phillip Isola. Straightening out the straight-through estimator: Overcoming optimization challenges in vector quantized networks. In *International Conference on Machine Learning*, pages 14096–14113. PMLR, 2023. 2, 3

[17] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*. Minneapolis, Minnesota, 2019. 6

[18] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[19] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11523–11532, 2022. 2, 3, 4, 7

[20] Tianhong Li, Huiwen Chang, Shlok Mishra, Han Zhang, Dina Katabi, and Dilip Krishnan. Mage: Masked generative encoder to unify representation learning and image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2142–2152, 2023. 7

[21] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *arXiv preprint arXiv:2406.11838*, 2024. 2, 3, 6, 7

[22] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 2

[23] Ilya Loshchilov, Cheng-Ping Hsieh, Simeng Sun, and Boris Ginsburg. ngpt: Normalized transformer with representation learning on the hypersphere. *arXiv preprint arXiv:2410.01131*, 2024. 5

[24] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022. 7

[25] Zhuoyan Luo, Fengyuan Shi, Yixiao Ge, Yujiu Yang, Limin Wang, and Ying Shan. Open-magvit2: An open-source project toward democratizing auto-regressive visual generation. *arXiv preprint arXiv:2409.04410*, 2024. 2, 3

[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*. 2, 3, 7

[27] Minheng Ni, Xiaoming Li, and Wangmeng Zuo. Nuwa-lip: language-guided image inpainting with defect-free vqgan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14183–14192, 2023. 7

[28] Zanlin Ni, Yulin Wang, Renping Zhou, Yizeng Han, Jiayi Guo, Zhiyuan Liu, Yuan Yao, and Gao Huang. Enat: Rethinking spatial-temporal interactions in token-based image synthesis. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 1, 7

[29] Zanlin Ni, Yulin Wang, Renping Zhou, Jiayi Guo, Jinyi Hu, Zhiyuan Liu, Shiji Song, Yuan Yao, and Gao Huang. Revisiting non-autoregressive transformers for efficient image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7007–7016, 2024. 1, 2

[30] Zanlin Ni, Yulin Wang, Renping Zhou, Rui Lu, Jiayi Guo, Jinyi Hu, Zhiyuan Liu, Yuan Yao, and Gao Huang. Adanat: Exploring adaptive policy for token-based image generation. In *ECCV*, 2024. 1

[31] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 1, 2, 7

[32] 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. In *The Twelfth International Conference on Learning Representations*. 1, 2

[33] Alec Radford. Improving language understanding by generative pre-training. 2018. 2

[34] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. 2

[35] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020. 1

[36] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019. 2

[37] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 2, 7

[38] Axel Sauer, Tero Karras, Samuli Laine, Andreas Geiger, and Timo Aila. Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis. In *International conference on machine learning*, pages 30105–30118. PMLR, 2023. 6

[39] Fengyuan Shi, Zhuoyan Luo, Yixiao Ge, Yujiu Yang, Ying Shan, and Limin Wang. Taming scalable visual tokenizer for autoregressive image generation. *arXiv preprint arXiv:2412.02692*, 2024. 3, 6

[40] Jie Shi, Chenfei Wu, Jian Liang, Xiang Liu, and Nan Duan. Divae: Photorealistic images synthesis with denoising diffusion decoder. *arXiv preprint arXiv:2206.00386*, 2022. 7

[41] 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. 1, 2, 3, 7

[42] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv preprint arXiv:2404.02905*, 2024. 1, 2, 6, 7

[43] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2

[44] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 2

[45] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 1

[46] Junke Wang, Yi Jiang, Zehuan Yuan, Binyue Peng, Zuxuan Wu, and Yu-Gang Jiang. Omnitokenizer: A joint image-video tokenizer for visual generation. *arXiv preprint arXiv:2406.09399*, 2024. 7

[47] 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, 2

[48] Ze Wang, Jiang Wang, Zicheng Liu, and Qiang Qiu. Binary latent diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22576–22585, 2023. 7

[49] Mark Weber, Lijun Yu, Qihang Yu, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. Maskbit: Embedding-free image generation via bit tokens. *Transactions on Machine Learning Research*. 2, 3, 7

[50] 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. 1

[51] Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. ivideogpt: Interactive videogpts are scalable world models. *arXiv preprint arXiv:2405.15223*, 2024. 2

[52] Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, et al. Vila-u: a unified foundation model integrating visual understanding and generation. *arXiv preprint arXiv:2409.04429*, 2024. 1

[53] 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. 1, 2

[54] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge,

and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021. 2, 3, 7

[55] Lijun Yu, Jose Lezama, Nitesh Bharadwaj 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. In *The Twelfth International Conference on Learning Representations*. 2, 3

[56] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Randomized autoregressive visual generation. *arXiv preprint arXiv:2411.00776*, 2024. 2

[57] Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. *arXiv preprint arXiv:2406.07550*, 2024. 2, 3

[58] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019. 5

[59] 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*, pages 586–595, 2018. 6

[60] Lei Zhu, Fangyun Wei, Yanye Lu, and Dong Chen. Scaling the codebook size of vqgan to 100,000 with a utilization rate of 99%. *arXiv preprint arXiv:2406.11837*, 2024. 2, 3, 7

[61] Yongxin Zhu, Bocheng Li, Hang Zhang, Xin Li, Linli Xu, and Lidong Bing. Stabilize the latent space for image autoregressive modeling: A unified perspective. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 2, 3