Towards Accurate Image Coding:
Improved Autoregressive Image Generation with Dynamic Vector Quantization

Mengqi Huang1, Zhendong Mao1,2*, Zhuowei Chen1, Yongdong Zhang1,2
1University of Science and Technology of China, Hefei, China; 2Institute of Artificial intelligence, Hefei Comprehensive National Science Center, Hefei, China
{huangmq, chenzw01}@mail.ustc.edu.cn, {zdmao, zhyd73}@ustc.edu.cn

Abstract

Existing vector quantization (VQ) based autoregressive models follow a two-stage generation paradigm that first learns a codebook to encode images as discrete codes, and then completes generation based on the learned codebook. However, they encode fixed-size image regions into fixed-length codes and ignore their naturally different information densities, which results in insufficiency in important regions and redundancy in unimportant ones, and finally degrades the generation quality and speed. Moreover, the fixed-length coding leads to an unnatural raster-scan autoregressive generation. To address the problem, we propose a novel two-stage framework: (1) Dynamic-Quantization VAE (DQ-VAE) which encodes image regions into variable-length codes based on their information densities for an accurate & compact code representation. (2) DQ-Transformer which thereby generates images autoregressively from coarse-grained (smooth regions with fewer codes) to fine-grained (details regions with more codes) by modeling the position and content of codes in each granularity alternately, through a novel stacked-transformer architecture and shared-content, non-shared position input layers designs. Comprehensive experiments on various generation tasks validate our superiorities in both effectiveness and efficiency. Code will be released at https://github.com/CrossmodalGroup/DynamicVectorQuantization.

1. Introduction

The vision community has witnessed the rapid progress of deep generative models, pushing image generation quality to an unprecedented level. As a fundamental task, generating realistic images from arbitrary inputs (e.g., class labels) can empower humans to create rich and diverse visual content and bring numerous real-world applications. Unifying the realism of local details and the consistency of global structure is the eternal pursuit for all image generations.

Recently, vector quantization (VQ) [37] has been a foundation for various types of generative models as evidenced by numerous large-scale diffusion models like LDM [32], autoregressive models like DALL-E [30], etc. These models follow a two-stage generation paradigm, i.e., the first stage learns a codebook by VQ to encode images as dis-
crete codes, where each code represents a local visual pattern, while the second stage learns to generate codes of local regions and then restores to images. The importance lies in that the local details could be well encoded in the first stage and thus the second stage could effectively focus on global structure modeling, leading to better generation quality and scalability. Existing models mainly focus on the second stage to better generate codes for improving generation quality, such as raster-scan autoregression [11, 30, 43], bi-direction [7, 24, 44], or diffusion [5, 14, 32]. Only a few works aim to improve the fundamental code representation itself in the first stage, including perceptual and adversarial loss for context-rich codebook [13], residual quantization [23], and more expressive transformer backbone [42], etc. Their commonality is that they all focus on encoding more information of all image regions together.

However, existing fundamental encoding works inherently fail to effectively encode image information for an accurate and compact code representation, because they ignore the naturally different information densities of different image regions and encode fixed-size regions into fixed-length codes. As a result, they suffer from two limitations: (1) insufficient coding for important regions with dense information, which fails to encode all necessary information for faithful reconstruction and therefore degrades the realism of local details in both stages. (2) Redundant coding for unimportant ones with sparse information, bringing huge redundant codes that mislead the second stage to focus on the redundancy and therefore significantly hinder the global structure modeling on important ones. As shown in Figure 1(a), the fixed-length codes result in large reconstruction errors in important cheetah regions and produce poor local details (e.g., face, hair) in both stages. Meanwhile, the fixed-length codes are overwhelmed for unimportant background regions, which misleads the second stage to generate redundant background and inconsistent cheetah structure. Moreover, as shown in Figure 1(b), since all regions are encoded into fixed-length codes, there is no way for the second stage to distinguish their varying importance and thus results in an unnatural raster-scan order [13] for existing autoregressive models [11, 23, 30, 42, 43], which fails to consider the image content for an effective generation.

To address this problem, inspired by the classical information coding theorems [18, 33, 34] and their dynamic coding principle, we propose information-density-based variable-length coding for an accurate and compact code representation to improve generation quality and speed. Moreover, we further propose a natural coarse-to-fine autoregressive model for a more effective generation. Specifically, we propose a novel two-stage generation framework: (1) Dynamic-Quantization VAE (DQ-VAE) which first constructs hierarchical image representations of multiple candidate granularities for each region, and then uses a novel Dynamic Grained Coding module to assign the most suitable granularity for each region under the constraint of a proposed budget loss, matching the percentage of each granularity to the desired expectation holistically. (2) DQ-Transformer which thereby generates images autoregressively from coarse-grained (smooth regions with fewer codes) to fine-grained (details regions with more codes) to more effectively achieve consistent structures. Considering the distribution of different granularities varying, DQ-Transformer models the position and content of codes in each granularity alternately through a novel stacked-transformer architecture. To effectively teach the difference between different granularities, we further design shared-content and non-shared-position input layers.

Our main contributions are summarized as follows:

Conceptual contribution. We point to the inherent insufficiency and redundancy in existing fixed-length coding since they ignore information density. For the first time, we propose information-density-based variable-length coding for accurate & compact code representations.

Technical contribution. (1) We propose DQ-VAE to dynamically assign variable-length codes to regions based on their different information densities through a novel Dynamic Grained Coding module and budget loss. (2) We propose DQ-Transformer to generate images autoregressively from coarse-grained to fine-grained for the first time, which models the position and content of codes alternately in each granularity by stacked-transformer architecture with shared-content and non-shared position input layers design.

Experimental contribution. Comprehensive experiments on various generations validate our superiority, e.g., we achieve 7.4% quality improvement and faster speed compared to existing state-of-the-art autoregressive model on unconditional generation, and 17.3% quality improvement compared to existing million-level parameters state-of-the-art models on class-conditional generation.

2. Related Works

2.1. Vector Quantization for Image Generation

Existing VQ-based models follow a two-stage paradigm that first learns a codebook to encode images into discrete space and then models the underlying distribution in this discrete space. The VQ-based paradigm has attracted increasing interest and is adopted by most milestone generative models, such as latent diffusion [32], DALL-E [30], Parti [43], etc. Most works focus on the second stage for better learning in the discrete space, such as discrete diffusion [1, 14, 32, 36, 45], bidirection [5, 7, 24, 44], and the most popular raster-scan autoregression [11, 13, 14, 23, 30, 31, 37, 43]. Only a few works aim to improve the fundamental encoding, e.g., VQGAN [13] introduces perceptual and adversarial loss for a context-rich codebook. [23] intro-
2.2. Dynamic Network

Designing dynamic architectures is an effective approach for efficient deep learning and yields better representation power and generality [15]. Literally, current research can be mainly categorized into three directions, i.e., dynamic depth for network early exiting [4] or layer skipping [39], dynamic width for skipping neurons [3] or channels [26] and dynamic routing for multi-branch structure networks [17,25,35,41]. Our work belongs in the last direction. To the best of our knowledge, the dynamic network has never been studied in VQ-based generation and we present the first work to realize the variable-length coding of classical information coding theorems through the dynamic network.

3. Methodology

Our overall two-stage framework is depicted in Figure 2. In the following, we will first briefly revisit the formulation of VQ and then describe our proposed method in detail.

3.1. Preliminary

Vector Quantization (VQ) [37] denotes the technique that learns a codebook to encode images into discrete code representations. Formally, the codebook is defined as $\mathcal{C} := \{ (k, e(k)) \}_{k \in [K]}$, where $K$ is the codebook size and $n_z$ is the dimension of codes. An image $\mathbf{X} \in \mathbb{R}^{H_0 \times W_0 \times 3}$ is first encoded into grid features $\mathbf{Z} = E(\mathbf{X}) \in \mathbb{R}^{H \times W \times n_z}$ by the encoder $E$, where $(H, W) = (H_0/f, W_0/f)$ and $f$ is the downsampling factor. For each vector $z \in \mathbb{R}^{n_z}$ in $\mathbf{Z}$, the quantization operation $Q(\cdot)$ replaces it with the code that has the closest euclidean distance with it in the codebook $\mathcal{C}$:

$$Q(z; \mathcal{C}) = \arg\min_{k \in [K]} ||z - e_k||_2.$$  (1)

Here, $Q(z; \mathcal{C})$ is the quantized code. $z^q = e(Q(z; \mathcal{C}))$ is the quantized vector. Therefore, the quantized encoded fea-
tures are $\mathbf{Z}^q \in \mathbb{R}^{H \times W \times n_s}$. The decoder $D$ is used to reconstruct the original image by $\hat{X} = D(\mathbf{Z}^q)$. Here each code roughly represents a fixed $f^2$ size visual pattern and each image region is represented by the same length of codes without distinguishing their different information densities. As a result, existing works suffer from both insufficiency in important regions and redundancy in unimportant ones.

3.2. Stage 1: Dynamic-Quantization VAE (DQ-VAE)

Different from existing works that adopt a fixed downsampling factor $f$ to represent image regions as fixed-length codes, DQ-VAE first defines a set of candidates:

$$ F = \{f_1, f_2, ..., f_K\}, $$

where $f_1 < f_2 < ... < f_K$, (2)

and encodes images into hierarchical features $\mathbf{Z} = \{\mathbf{Z}_1, \mathbf{Z}_2, ..., \mathbf{Z}_K\}$ through a hierarchical encoder $E_h$, where $\mathbf{Z}_i \in \mathbb{R}^{H_i \times W_i \times n_s}$ and $(H_i, W_i) = (H_0/f_i, W_0/f_i)$ for each $i \in \{1, 2, ..., K\}$. The image region’s size is set as the maximum downsampling factor, i.e., $S = f_K$, and therefore each $S^2$ size image region now has multiple granularity representations containing different numbers of features. Then the Dynamic Grained Coding (DGC) module assigns the most suitable granularity for each region and results in multi-grained representations, which are further quantized by VQ. To deal with the irregular code map that different regions have different numbers of codes, we further propose a simple but effective nearest-neighbor replication, that is, in each region the quantized codes are replicated to the code number of the finest granularity if the finest granularity is not assigned for it, resulting in a regular code map that could be conveniently decoded by the convolutional decoder $D$.

Dynamic Grained Coding (DGC). As illustrated in Figure 3, given the encoded hierarchical image features $\mathbf{Z} = \{\mathbf{Z}_1, \mathbf{Z}_2, ..., \mathbf{Z}_K\}$, we implement a discrete gating network with Gumbel-Softmax technique [19] to determine the granularity for each image region. Specifically, each granularity feature is first normed by group-normalization to stabilize training and then pooled to the size of the coarsest granularity feature by average-pooling, except the coarsest granularity (i.e., $f_K$) feature itself. The pooled features are denoted as $\mathbf{Z}' = \{\mathbf{Z}'_1, \mathbf{Z}'_2, ..., \mathbf{Z}'_K\}$ and $\mathbf{Z}'_i \in \mathbb{R}^{H_i' \times W_i' \times n_s}$ for $i \in \{1, 2, ..., K\}$, where $(H_i', W_i') = (H_0/f_K, W_0/f_K)$. The gating logits $\mathbf{G}$ are generated as:

$$ \mathbf{G} = (\mathbf{Z}'_1 || \mathbf{Z}'_2 || ... || \mathbf{Z}'_K) \mathbf{W}_g \in \mathbb{R}^{H \times W \times K}, $$

where $\|$ is the concatenation operation along the channel dimension and $\mathbf{W}_g \in \mathbb{R}^{K \times n_s \times K}$ is the learnable weight.

For each region $(i, j)$, its gating logits $g_{i,j} \in \mathbb{R}^K$ is used to decide the granularity by calculating the gating index:

$$ \theta_{i,j} = \arg \max_k (g_{i,j,k}) \in \{1, 2, ..., K\}. $$

To enable the end-to-end training of this discrete process, inspired by [40, 46], the determined decisions in Eq. (4) are replaced with the stochastic sampling process. Mathematically, given a categorical distribution with unnormalized log probabilities, discrete gating indices can be yielded with noise samples drawn from a standard Gumbel distribution:

$$ \theta_{i,j} = \arg \max_k (g_{i,j,k} + n_k), \text{ where } n_k \sim \text{Gumbel}(0,1). \quad (5) $$

To enable the back-propagation of the above hard decision, we adopt the Gumbel-Softmax technique [19] to give a continuous and differentiable approximation by replacing the argmax with a softmax operation. The soft gating score $p_{i,j}$ for each region is then selected by the gating indices:

$$ p_{i,j} = \frac{\exp((g_{i,j,\theta_{i,j}} + n_{\theta_{i,j}})/\tau)}{\sum_{k}^{K} \exp((g_{i,j,k} + n_k)/\tau)} \in [0, 1], $$

where the temperature $\tau = 1$. We use a straight-through estimator for the gradients of gating logits, which are obtained through the soft gating score $p_{i,j}$ during the backward pass. The above stochastic process is only adopted during training and no random sampling is required during inference.

**Budget Loss.** We adopt the training loss of VQGAN [13] as $L_{\text{vanilla}}$, which includes reconstruction loss ($l_1$ loss, perceptual loss, adversarial loss) and quantization loss. In the absence of a budget constraint, the DGC module typically prefers to assign the finest granularity for all image regions, which is in contrast to our purpose. Therefore, we further propose a budget loss to match the percentage of each granularity to our desired expectation. Specifically, we denote the desired ratio of each granularity $k$ as $r_k$ and $\sum_k r_k = 1$. For an image sample whose current assigned ratio of each granularity $k$ is $r'_k$, we define budget loss as:

$$ L_{\text{budget}} = \sum_k^{K} (r_k - r'_k)^2, $$

where we only calculate on $K - 1$ granularities since the ratio of the last granularity is determined by $1 - \sum_k^{K-1} r_k$.

The final loss for DQ-VAE is defined as:

$$ L_{\text{stage1}} = L_{\text{vanilla}} + \lambda L_{\text{budget}}, $$

where $\lambda$ is the budget loss weight.
where $\lambda$ is a loss balance hyper-parameter. The expected ratio of each granularity is holistic on the dataset level. Therefore, since important regions contribute more to the reconstruction quality, the variable-length coding is realized from two aspects, i.e., inter-dynamic, longer code sequence for complex images while shorter code sequence for easy ones; intra-dynamic, for each image, more codes for important regions while fewer codes for unimportant ones.

### 3.3. Stage 2: DQ-Transformer

Different images share different perceptually important regions and different complexities. Therefore, DQ-VAE encodes images as the code sequence of variable lengths and the distribution of each granularity region in images is also completely different. Though learning this dynamic underlying prior is very challenging, it also opens a promising potential for autoregressive image generation, that is, a natural and more effective coarse-grained to fine-grained generation order since DQ-VAE naturally divides coarse regions (smooth regions with fewer codes) apart from fine regions (details regions with more codes). Imagine image generation as a jigsaw puzzle problem, it is more effective and efficient that we first fill in the large and easy pieces (coarse regions) and then fill in the small and difficult ones (fine regions). With this motivation, DQ-Transformer first constructs the codes’ content and position sequence in each granularity separately and then concatenates them in a coarse-to-fine manner to autoregressively predict the next code’s position and content through the stacked Position-Transformer and Content-Transformer. The distinction of different granularities is realized by the shared content, non-shared-position, and granularity input layers designs.

**Training sequence construction.** As illustrated in stage 2(a) in Figure 2, the sequence of each granularity is constructed separately. As for the content sequence, each index is the quantized code index. As for the position sequence, each index is the position of the corresponding code index in the position map of current granularity. We add a special $<\text{sos}>$ code at the beginning of all content and position sequences to indicate the start of the sequence, and another special $<\text{eos}>$ code at the end of them to indicate the end of the sequence. To enable batch training and sampling, we use a special $<\text{pad}>$ code to pad all samples to the same length in each granularity. Finally, we concatenate all granularities’ content and position sequences in a coarse-to-fine manner, which we denote as $C$ and $P$, respectively.

**Position-Transformer.** We first learn to predict the next code position conditioned on all previous codes and their positions. The input of Position-Transformer consists of four parts: (1) content embedding which is calculated from $C$ by a shared-content layer for all granularities, (2) position embedding which is calculated from $P$ by non-shared-position layers for each granularity separately, (3) granularity embedding which is used for distinguishing each granularity, and (4) a learned absolute position embedding for making the network aware of the absolute position of the sequence, which is the same as most transformer-architecture [13, 36, 38]. After processing by Position-Transformer, the output hidden vector $H_p$ encodes both code and their position information and is used for next position predicting. The negative log-likelihood (NLL) loss for the next code position autoregressive training is:

$$
\mathcal{L}_{\text{position}} = \mathbb{E}( -\log p(P|P_{<l}, C_{<l}))
$$

#### Content-Transformer. We then learn to predict the next code’s content conditioned on all previous codes and the position of current code. Specifically, The input of Content-Transformer is two parts: (1) the output of Position-Transformer $H_p$ and (2) the ground-truth information of the current position which also is calculated by the non-shared-position layers. For example, if the input position sequence for Position-Transformer is $P_{[0:2]}$, then the input ground-truth position sequence for Content-Transformer is $P_{[1:2]}$. The negative log-likelihood (NLL) loss for the next code’s content autoregressive training is:

$$
\mathcal{L}_{\text{content}} = \mathbb{E}( -\log p(C|P_{\leq l}, C_{<l}))
$$

**Training & Inference.** During training, the total loss for DQ-Transformer is defined as:

$$
\mathcal{L}_{\text{stage2}} = \mathcal{L}_{\text{position}} + \mathcal{L}_{\text{content}}.
$$

Our proposed DQ-Transformer is a general visual generative model which could be easily extended to various other generation tasks. As for the class-conditional generation, we replace the $<\text{sos}>$ code in the content sequence of each granularity with the class-label code. During inference, we could also autoregressively generate images from coarse-grained to fine-grained, as illustrated in Algorithm 1, where we take the unconditional generation as an example and other conditional generations can be derived accordingly.

### 4. Experiments

**Benchmarks.** We evaluate our method on unconditional FFHQ [20] benchmark and class-conditional ImageNet [9] benchmark with $256 \times 256$ image resolution.

**Metrics.** Following previous works [13, 23, 44], the standard Fréchet Inception Distance (FID) [16] is adopted for evaluating the generation and reconstruction quality (denoted as rFID). rFID is calculated over the entire test set. Inception Score (IS) [16] is also adopted for class-conditional generation on the ImageNet benchmark.

**Implementation.** DQ-VAE follows the architecture of VQGAN [13] except for the lightweight DRC module, which is trained with the codebook size $K = 1024$ and $\lambda =$
Figure 4. Qualitative results. Left: Our unconditional generation on FFHQ. Right: Our class-conditional generation on ImageNet.

<table>
<thead>
<tr>
<th>Model</th>
<th>$L$</th>
<th>#Params</th>
<th>FID↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>VQGAN$^{(21)}$ [13]</td>
<td>256</td>
<td>(72.1+307)M</td>
<td>11.4</td>
</tr>
<tr>
<td>DCT$^{(21)}$ [28]</td>
<td>&gt;1024</td>
<td>738M</td>
<td>13.06</td>
</tr>
<tr>
<td>ViT-VQGAN$^{(22)}$ [42]</td>
<td>1024</td>
<td>(599+1697)M</td>
<td>5.3</td>
</tr>
<tr>
<td>RQ-VAE$^{(22)}$ [23]</td>
<td>256</td>
<td>(100+355)M</td>
<td>10.38</td>
</tr>
<tr>
<td>Mo-VQGAN$^{(22)}$ [44]</td>
<td>1024</td>
<td>(82.7+307)M</td>
<td>8.52</td>
</tr>
<tr>
<td>DQ-Transformer$_b$</td>
<td>640</td>
<td>(47.5+308)M</td>
<td>4.91</td>
</tr>
</tbody>
</table>

Table 1. Comparison of unconditional autoregressive generation on FFHQ. $L$ is coding length. #Params splits in (VAE+autoregressive model).

<table>
<thead>
<tr>
<th>Type</th>
<th>Model</th>
<th>$L$</th>
<th>#Params</th>
<th>IS↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAN</td>
<td>BigGAN-deep [6]</td>
<td>-</td>
<td>160M</td>
<td>198.2</td>
</tr>
<tr>
<td>diffusion</td>
<td>[29]</td>
<td>-</td>
<td>280M</td>
<td>12.26</td>
</tr>
<tr>
<td>diffusion</td>
<td>ADM [10]</td>
<td>-</td>
<td>554M</td>
<td>101.0</td>
</tr>
<tr>
<td>bi-direct</td>
<td>MaskGIT [1]</td>
<td>1024</td>
<td>227M</td>
<td>182.1</td>
</tr>
<tr>
<td>ARM</td>
<td>VQGAN$^{*}$ [13]</td>
<td>256</td>
<td>379M</td>
<td>75.75</td>
</tr>
<tr>
<td>ARM</td>
<td>DCT [28]</td>
<td>&gt;1024</td>
<td>738M</td>
<td>-</td>
</tr>
<tr>
<td>ARM</td>
<td>RQ-VAE [23]</td>
<td>256</td>
<td>480M</td>
<td>86.8</td>
</tr>
<tr>
<td>ARM</td>
<td>RQ-VAE [23]</td>
<td>256</td>
<td>821M</td>
<td>104.3</td>
</tr>
<tr>
<td>ARM</td>
<td>Mo-VQGAN [44]</td>
<td>1024</td>
<td>389M</td>
<td>138.3</td>
</tr>
<tr>
<td>ARM</td>
<td>DQ-Transformer$_b$</td>
<td>640</td>
<td>355M</td>
<td>152.8</td>
</tr>
<tr>
<td>ARM</td>
<td>DQ-Transformer$_l$</td>
<td>640</td>
<td>655M</td>
<td>178.2</td>
</tr>
</tbody>
</table>

Table 2. Comparison of class-conditional generation with million-level parameters on ImageNet. $L$ is coding length. ARM denotes for autoregressive model. * denotes for our reproduction.

10. DQ-Transformer adopts a stack of causal self-attention blocks [38] and is trained with two different settings, i.e., DQ-Transformer$_b$ (base) with 6 layers Position-Transformer and 18 layers Content-Transformer of a total 308M parameters, and DQ-Transformer$_l$ (large) with 6 layers Position-Transformer and 42 layers Content-Transformer of a total 608M parameters to demonstrate our scalability. All models are trained with eight RTX-3090 GPUs. Top-k and top-p sampling are used to report the best performance. More details can be found in the supplementary.

4.1. Comparison with state-of-the-art methods

The main results are reported on dual granularities of $F = \{8, 16\}$, and the ratio $r_{f=8} = 0.5$ (640 average length).

Unconditional generation. As shown in Table 1, our model outperforms all existing autoregressive state-of-the-art models including the strongest large-scale ViT-VQGAN [42] by a 7.4% quality improvement. We compare with other types of state-of-the-art models in Table 4 and also achieve top-level performance. The qualitative results of unconditional generation are shown on the left of Figure 4.

Class-conditional generation. The comparison is split into Million/Billion according to whether they can be trained under normal computing resources (i.e., 24G memory 3090). We first compare with all million-level parameters state-of-the-art in Table 2. Our model with 355M parameters already outperforms all autoregressive and diffusion models. Moreover, our model with 655M outperforms GAN-based and bi-direct state-of-the-art, which demonstrates our effectiveness and scalability. We further compare with large-scale billion-level state-of-the-art in Table 3, where we achieve top-level performance with fewer parameters. The qualitative results of class-conditional generation are shown on the right of Figure 4.

4.2. Ablations & Analysis

Analysis on DQ-VAE. We first demonstrate that our variable-length coding has better reconstruction compared to the existing fixed-length one in Table 5. We take VQ-
Figure 5. Visualization of the variable-length coding of our DQ-VAE, where our coding map exactly matches the error map of VQGAN and therefore leads to better reconstruction quality, i.e., the information-dense regions where VQGAN has higher reconstruction error are assigned to more codes, while information-sparse regions where VQGAN has lower reconstruction error are assigned to few codes.

Table 3. Comparison between our million-level model and large-scale billion-level big models of class-conditional generation on ImageNet.

<table>
<thead>
<tr>
<th>Type</th>
<th>Model</th>
<th>$L$</th>
<th>#Params</th>
<th>FID↓</th>
<th>IS↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diffusion</td>
<td>ImageBART [12]</td>
<td>-</td>
<td>3.5B</td>
<td>21.19</td>
<td>61.6</td>
</tr>
<tr>
<td>ARM</td>
<td>VQ-VAE-2 [31]</td>
<td>5120</td>
<td>13.5B</td>
<td>31.11</td>
<td>45</td>
</tr>
<tr>
<td>ARM</td>
<td>VQGAN [13]</td>
<td>256</td>
<td>1.4B</td>
<td>15.78</td>
<td>78.3</td>
</tr>
<tr>
<td>ARM</td>
<td>VIT-VQGAN [36]</td>
<td>1024</td>
<td>2.2B</td>
<td>4.17</td>
<td>175.1</td>
</tr>
<tr>
<td>ARM</td>
<td>RQ-VAE [33]</td>
<td>256</td>
<td>3.8B</td>
<td>7.55</td>
<td>134</td>
</tr>
<tr>
<td>ARM</td>
<td>DQ-Transformerb</td>
<td>640</td>
<td>355M</td>
<td>7.34</td>
<td>152.3</td>
</tr>
<tr>
<td>ARM</td>
<td>DQ-Transformerl</td>
<td>640</td>
<td>655M</td>
<td>5.11</td>
<td>178.2</td>
</tr>
</tbody>
</table>

Table 4. Comparison with other types of state-of-the-art on unconditional FFHQ, where we further improve the quality of autoregressive models.

<table>
<thead>
<tr>
<th>Model Type</th>
<th>Model</th>
<th>FID↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAN</td>
<td>BigGAN [6]</td>
<td>12.4</td>
</tr>
<tr>
<td>GAN</td>
<td>StyleGAN2 [21]</td>
<td>3.8</td>
</tr>
<tr>
<td>VAE</td>
<td>VDVAE [8]</td>
<td>28.5</td>
</tr>
<tr>
<td>Diffusion</td>
<td>ImageBART [12]</td>
<td>9.57</td>
</tr>
<tr>
<td>Diffusion</td>
<td>UDM [22]</td>
<td>5.54</td>
</tr>
<tr>
<td>Autoregressive</td>
<td>DQ-Transformerb</td>
<td>4.91</td>
</tr>
</tbody>
</table>

Table 5. Ablations of the proposed variable-length coding on ImageNet. Here $F$ denotes the granularity candidates set. “ratio” denotes the ratio of each granularity. We show that variable-length coding could bring better reconstruction compared to fixed-length coding on the same code length.

<table>
<thead>
<tr>
<th>Model</th>
<th>$F$</th>
<th>ratio</th>
<th>rFID↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>VQGAN [13]</td>
<td>(8,16,32)</td>
<td>{0.05, 0.75, 0.3}</td>
<td>4.82</td>
</tr>
<tr>
<td>DQ-VAE</td>
<td>(8,16,32)</td>
<td>{0.075, 0.625, 0.3}</td>
<td>4.08</td>
</tr>
<tr>
<td>DQ-VAE / random</td>
<td>(8,16,32)</td>
<td>{0.075, 0.625, 0.3}</td>
<td>7.32</td>
</tr>
<tr>
<td>DQ-VAE</td>
<td>(8,16,32)</td>
<td>{0.1, 0.5, 0.4}</td>
<td>4.96</td>
</tr>
<tr>
<td>DQ-VAE</td>
<td>(8,16,32)</td>
<td>{0.125, 0.375, 0.5}</td>
<td>6.39</td>
</tr>
</tbody>
</table>

GAN [13] of $f = 16$ as the baseline, and DQ-VAE adopts triple granularities of $F = \{8, 16, 32\}$ and subject to:

$$r_{f=32} = 4 \times r_{f=8},$$  \hspace{1cm} (12)

which ensures DQ-VAE’s expected mean code length is the same as VQGAN (i.e., 256). We could conclude: (1) With a proper ratio, DQ-VAE’s variable-length coding achieves better reconstruction quality compared to VQGAN’s fixed-length one (ours 4.08 vs. VQGAN’s 4.82). The reason is that important regions require more codes to encode necessary information, while fewer codes are enough for unimportant ones since they are less informative. The phenomenon also reveals that existing fixed-length coding is both insufficient in important regions and redundant in unimportant ones. (2) When we improperly increase $r_{f=8}$, we get a larger $r_{f=32}$ which will inevitably assign some important regions with fewer codes and thus degrade the reconstruction quality. (3) Moreover, DQ-VAE’s adaptive assignment significantly outperforms the random one (ours 4.08 vs. random’s 7.32) which demonstrates that DQ-VAE could distinguish important regions from unimportant ones.

We then analyze the impact of different ratio percentages in Table 6, where DQ-VAE adopts dual granularities of $F = \{8, 16\}$. We show that: (1) The mean code length of each ratio matches the expectation well, which validates our proposed budget loss. (2) The results are consistent with the Pareto principle, which is also known as the 80/20 law. To be specific, when increasing $r_{f=8}$ from 0 to 0.3, we get 1.44 FID improvement while only a slight codebook usage drop, which indicates that the first 30% percentage important regions contribute the most valid information of images and existing fixed-length coding is insufficient in them. Mean-
while, when increasing $r_{f=8}$ from 0.7 to 1.0, we only get a subtle 0.21 FID improvement but a significant 9.6% codebook usage drop, which indicates that the last 30% percentage unimportant regions contribute little valid information of images but most redundancy. The experimental results strongly support our motivations for variable-length coding to get rid of insufficiency and redundancy simultaneously.

We visualize our variable-length coding on ImageNet in Figure 5, where DQ-VAE adopts dual granularities of $F = \{8, 16\}$ and $r_{f=8} = 0.3$. The error map is calculated by $l_1$ loss of each $16^2$ size region between images and VQGAN ($f = 16$) reconstructions. The red regions in our coding map are assigned to $f = 8$ (4 codes) while the blue ones are assigned to $f = 16$ (1 code). We show that our coding map matches VQGAN’s error map, i.e., important regions are assigned to more codes and unimportant ones are assigned to few codes, leading to better reconstruction quality.

**Analysis on the effectiveness of DQ-Transformer.** We first validate our input layers designs in Table 7. The non-shared-position and granularity layers are very important since they distinguish different granularities. Without these designs, DQ-Transformer fails to know which granularity of code should be generated next, and thus performs worse.

We then analyze the generation quality of different ratios in Figure 6 left. The generation speed of autoregressive models mostly depends on their code length. The Pareto curve shows that the generation quality (FID) saturates when $r_{f=8}$ reaches 0.5. The experimental phenomenon reveals that a proper ratio is important for the unity of a high generation quality and fast generation speed since it guarantees effective coding in both important regions and unimportant ones for an accurate & compact code representation.

**Analysis on the efficiency of DQ-Transformer.** We compare our generation speed to the existing state-of-the-art autoregressive model ViT-VQGAN [42] according to different batch sizes in Figure 6 right. The generation speeds are evaluated on a single RTX-3090 GPU and the setup of ViT-VQGAN is implemented the same as its original paper. Our model achieves a much faster generation speed for all batch sizes which validates the efficiency brought by our accurate and compact code representation.

## 5. Conclusion & Future Direction
In this study, we point out that the existing fixed-length coding ignores the naturally different information densities of image regions and is inherently limited by insufficiency and redundancy, which degrades generation quality and speed. Moreover, the fixed-length coding brings an unnatural raster-scan autoregression. We thereby propose a novel two-stage generation framework: (1) DQ-VAE which dynamically assigns variable-length codes to regions based on their information densities for an accurate and compact code representation. (2) DQ-Transformer which then models the position and content of codes alternately, generating images autoregressively in a more natural and effective coarse-to-fine order for the first time. To effectively teach the difference between different granularities, we further design shared-content, non-shared-position, and granularity input layers. Comprehensive experiments on various image generations validate our effectiveness and efficiency.

**Future Direction.** VQ is the foundation for modern autoregressive [11, 23, 30, 43, 44], discrete diffusion [14, 32], and bidirectional [7] generation, and even pretraining [2, 27]. Our study validates the effectiveness and efficiency of the variable-length coding for autoregressive generation, but its great potential for diffusion, bi-direction, and pretraining is worth further exploration in the future.

## 6. Acknowledgments
This work is supported by National Natural Science Foundation of China under Grant 62222212 and Science Fund for Creative Research Groups under Grant 62121002.
References


[27] Chengzhi Mao, Lu Jiang, Mostafa Dehghani, Carl Vondrick, Rahul Sukthankar, and Irfan Essa. Discrete representations


