Autoregressive Image Generation using Residual Quantization

Doyup Lee*
POSTECH, Kakao Brain
doyup.lee@postech.ac.kr

Chiheon Kim*
Kakao Brain
chiheon.kim@kakaobrain.com

Saehoon Kim
Kakao Brain
shkim@kakaobrain.com

Minsu Cho
POSTECH
mscho@postech.ac.kr

Wook-Shin Han †
POSTECH
wshan@postech.ac.kr

Abstract

For autoregressive (AR) modeling of high-resolution images, vector quantization (VQ) represents an image as a sequence of discrete codes. A short sequence length is important for an AR model to reduce its computational costs to consider long-range interactions of codes. However, we postulate that previous VQ cannot shorten the code sequence and generate high-fidelity images together in terms of the rate-distortion trade-off. In this study, we propose the two-stage framework, which consists of Residual-Quantized VAE (RQ-VAE) and RQ-Transformer, to effectively generate high-resolution images. Given a fixed codebook size, RQ-VAE can precisely approximate a feature map of an image and represent the image as a stacked map of discrete codes. Then, RQ-Transformer learns to predict the quantized feature vector at the next position by predicting the next stack of codes. Thanks to the precise approximation of RQ-VAE, we can represent a 256×256 image as 8×8 resolution of the feature map, and RQ-Transformer can efficiently reduce the computational costs. Consequently, our framework outperforms the existing AR models on various benchmarks of unconditional and conditional image generation. Our approach also has a significantly faster sampling speed than previous AR models to generate high-quality images.

Figure 1. Examples of our conditional generation for 256×256 images. The images in the first row are generated from the classes of ImageNet. The images in the second row are generated from text conditions ("A cheeseburger in front of a mountain range covered with snow." and "a cherry blossom tree on the blue ocean"). The text conditions are unseen during the training.

1. Introduction

Vector quantization (VQ) becomes a fundamental technique for autoregerssive (AR) models to generate high-resolution images [5, 11, 12, 33, 40]. Specifically, an image is represented as a sequence of discrete codes, after the feature map of the image is quantized by VQ and rearranged by an ordering such as raster-scan [30]. After the quantization, AR model is trained to sequentially predict the codes in the sequence. That is, AR models can generate high-resolution images without predicting whole pixels in an image.

We postulate that reducing the sequence length of codes is important for AR modeling of images. A short sequence of codes can significantly reduce the computational costs of an AR model, since an AR uses the codes in previous positions to predict the next code. However, previous studies have a limitation to reducing the sequence length of images in terms of the rate-distortion trade-off [38]. Namely, VQ-VAE [40] requires an exponentially increasing size of

*Equal contribution
†Corresponding author

codebook to reduce the resolution of the quantized feature map, while conserving the quality of reconstructed images. However, a huge codebook leads to the increase of model parameters and the codebook collapse problem [8], which makes the training of VQ-VAE unstable.

In this study, we propose a *Residual-Quantized VAE (RQ-VAE)*, which uses a residual quantization (RQ) to precisely approximate the feature map and reduce its spatial resolution. Instead of increasing the codebook size, RQ uses a fixed size of codebook to recursively quantize the feature map in a coarse-to-fine manner. After D iterations of RQ, the feature map is represented as a stacked map of D discrete codes. Since RQ can compose as many vectors as the codebook size to the power of D, RQ-VAE can precisely approximate a feature map, while conserving the information of the encoded image without a huge codebook. Thus, RQ-VAE can further reduce the spatial resolution of the quantized feature map than previous studies [12, 33, 40]. For example, our RQ-VAE can use 8×8 resolution of feature maps for AR modeling of 256×256 images.

In addition, We propose *RQ-Transformer* to predict the codes extracted by RQ-VAE. For the input of RQ-Transformer, the quantized feature map in RQ-VAE is converted into a sequence of feature vectors. Then, RQ-Transformer predicts the next D codes to estimate the feature vector at the next position. Thanks to the reduced resolution of feature maps by RQ-VAE, RQ-Transformer can significantly reduce the computational costs and easily learn the long-range interactions of inputs. We also propose two training techniques for RQ-Transformer, *soft labeling* and *stochastic sampling* for the codes of RQ-VAE. They further improve the performance by resolving the exposure bias [34] in the training of RQ-Transformer. Consequently, our model can generate high-quality images in Figure 1.

Our main contributions are summarized as follows. 1) We propose RQ-VAE, which represents an image as a stacked map of discrete codes, while producing high-fidelity reconstructed images. 2) We propose RQ-Transformer to effectively predict the codes of RQ-VAE and its training techniques to resolve the exposure bias. 3) We show that our approach outperforms previous AR models and significantly improves the quality of generated images, computational costs, and sampling speed.

2. Related Work

AR Modeling for Image Synthesis AR models have shown promising results of image generation [5, 11, 12, 29, 35, 40] as well as text [4] and audio [8] generation. AR modeling of raw pixels is possible [5, 30, 31, 36], but it is infeasible for high-resolution images due to the slow speed and low quality of generated images. Thus, previous studies incorporate VQ-VAE [12], which uses VQ to represent an image as discrete codes, and uses an AR model to predict the codes of VQ-VAE. VQ-GAN [12], improves the perceptual quality of reconstructed images using adversarial [14, 20] and perceptual loss [25]. However, when the resolution of the feature map is further reduced, VQ-GAN cannot precisely approximate the feature map of an image due to the limited size of codebook.

VQs in Other Applications Composite quantizations have been used in other applications to represent a vector as a composition of codes for the precise approximation under a given codebook size [1, 13, 15, 26–28]. For the nearest neighbor search, product quantization (PQ) [15] approximates a vector as the sum of linearly independent vectors in the codebook. As a generalized version of PQ, additive quantization (AQ) [1] uses the dependent vectors in the codebook, but finding the codes is an NP-hard task [6]. Residual quantization (RQ, also known as stacked quantization) [22, 28] iteratively quantizes a vector and its residuals to represent the vector as a stack of codes, which has been used for neural network compression [13, 26, 27]. Our RQ-VAE adopts RQ to discretize the feature map of an image for AR modeling of images and uses a single shared codebook for all quantization steps.

3. Methods

We propose the two-stage framework with *RQ-VAE* and *RQ-Transformer* for AR modeling of images (see Figure 2). RQ-VAE uses a codebook to represent an image as a stacked map of D discrete codes. Then, our RQ-Transformer autoregressively predicts the next D codes at the next spatial position. We also introduce how our RQ-Transformer resolves the exposure bias [34] in the training of AR models.

3.1. Stage 1: Residual-Quantized VAE

In this section, we first introduce the formulation of VQ and VQVAE. Then, we propose RQ-VAE, which can precisely approximate a feature map without increasing the codebook size, and explain how RQ-VAE represents an image as a stacked map of discrete codes.

3.1.1 Formulation of VQ and VQ-VAE

Let a *codebook* $\mathcal{C}$ be a finite set $\{(k, \mathbf{e}(k))\}_{k\in[K]}$, which consists of the pairs of a *code* k and its *code embedding* $\mathbf{e}(k) \in \mathbb{R}^{n_z}$, where K is the codebook size and n_z is the dimensionality of code embeddings. Given a vector $\mathbf{z} \in \mathbb{R}^{n_z}$, $\mathcal{Q}(\mathbf{z};\mathcal{C})$ denotes VQ of $\mathbf{z}$, which is the code whose embedding is nearest to z, that is,

$$\mathcal{Q}(\mathbf{z};\mathcal{C}) = \arg\min_{k\in[K]} \|\mathbf{z} - \mathbf{e}(k)\|_2^2. \quad (1)$$

After VQ-VAE encodes an image into a discrete code map, VQ-VAE reconstructs the original image from the en-

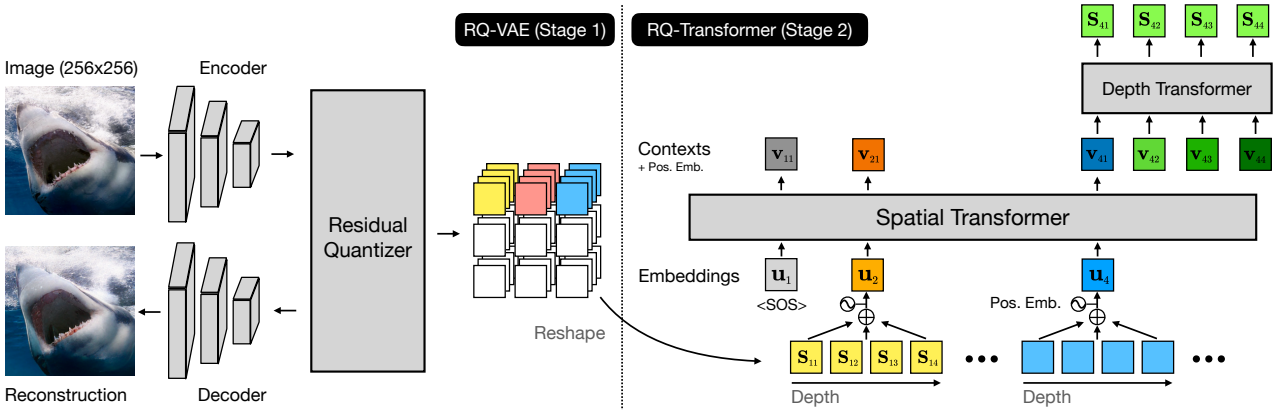

Figure 2. An overview of our two-stage image generation framework composed of RQ-VAE and RQ-Transformer. In stage 1, RQ-VAE uses the residual quantizer to represent an image as a stack of $D = 4$ codes. After the stacked map of codes is reshaped, RQ-Transformer predicts the D codes at the next position. More details are available in Section 3.

coded code map. Let E and G be an encoder and a decoder of VQ-VAE. Given an image $\mathbf{X} \in \mathbb{R}^{H_o \times W_o \times 3}$, VQ-VAE extracts the feature map $\mathbf{Z} = E(\mathbf{X}) \in \mathbb{R}^{H \times W \times n_z}$, where $(H, W) = (H_o/f, W_o/f)$ is the spatial resolution of $\mathbf{Z}$, and f is a downsampling factor. By applying the VQ to each feature vector at each position, VQ-VAE quantizes $\mathbf{Z}$ and returns its code map $\mathbf{M} \in [K]^{H \times W}$ and its quantized feature map $\hat{\mathbf{Z}} \in \mathbb{R}^{H \times W \times n_z}$ as

$$\mathbf{M}_{hw} = \mathcal{Q}(\mathbf{Z}_{hw}; \mathcal{C}), \quad \hat{\mathbf{Z}}_{hw} = \mathbf{e}(\mathbf{M}_{hw}), \tag{2}$$

where $\mathbf{Z}_{hw} \in \mathbb{R}^{n_z}$ is a feature vector at (h, w), and $\mathbf{M}_{hw}$ is its code. Finally, the input is reconstructed as $\hat{\mathbf{X}} = G(\hat{\mathbf{Z}})$.

We remark that reducing the spatial resolution of $\hat{\mathbf{Z}}$, (H, W), is important for AR modeling, since the computational cost of an AR model increases with HW. However, since VQ-VAE conducts a lossy compression of images, there is a trade-off between reducing (H, W) and conserving the information of $\mathbf{X}$. Specifically, VQ-VAE with the codebook size K uses $HW \log_2 K$ bits to represent an image as the codes. Note that the best achievable reconstruction error depends on the number of bits in terms of the rate-distortion theory [38]. Thus, to further reduce (H, W) to $(H/2, W/2)$ but preserve the reconstruction quality, VQ-VAE requires the codebook of size K^4. However, VQ-VAE with a large codebook is inefficient due to the codebook collapse problem [8] with unstable training.

3.1.2 Residual Quantization

Instead of increasing the codebook size, we adopt a residual quantization (RQ) to discretize a vector $\mathbf{z}$. Given a quantization depth D, RQ represents $\mathbf{z}$ as an *ordered* D codes

$$\mathcal{RQ}(\mathbf{z}; \mathcal{C}, D) = (k_1, \cdots, k_D) \in [K]^D, \tag{3}$$

where $\mathcal{C}$ is the codebook of size $|\mathcal{C}| = K$, and k_d is the code of $\mathbf{z}$ at depth d. Starting with 0-th residual $\mathbf{r}_0 = \mathbf{z}$, RQ recursively computes k_d, which is the code of the residual $\mathbf{r}_{d-1}$, and the next residual $\mathbf{r}_d$ as

$$\begin{aligned} k_d &= \mathcal{Q}(\mathbf{r}_{d-1}; \mathcal{C}), \\ \mathbf{r}_d &= \mathbf{r}_{d-1} - \mathbf{e}(k_d), \end{aligned} \tag{4}$$

for $d = 1, \cdots, D$. In addition, we define $\hat{\mathbf{z}}^{(d)} = \sum_{i=1}^{d} \mathbf{e}(k_i)$ as the partial sum of up to d code embeddings, and $\hat{\mathbf{z}} := \hat{\mathbf{z}}^{(D)}$ is the quantized vector of $\mathbf{z}$.

The recursive quantization of RQ approximates the vector $\mathbf{z}$ in a coarse-to-fine manner. Note that $\hat{\mathbf{z}}^{(1)}$ is the closest code embedding $\mathbf{e}(k_1)$ in the codebook to $\mathbf{z}$. Then, the remaining codes are subsequently chosen to reduce the quantization error at each depth. Hence, the partial sum up to d, $\hat{\mathbf{z}}^{(d)}$, provides a finer approximation as d increases.

Although we can separately construct a codebook for each depth d, a single shared codebook $\mathcal{C}$ is used for every quantization depth. The shared codebook has two advantages for RQ to approximate a vector $\mathbf{z}$. First, using separate codebooks requires an extensive hyperparameter search to determine the codebook size at each depth, but the shared codebook only requires to determine the total codebook size K. Second, the shared codebook makes all code embeddings available for every quantization depth. Thus, a code can be used at every depth to maximize its utility.

We remark that RQ can more precisely approximate a vector than VQ when their codebook sizes are the same. While VQ partitions the entire vector space $\mathbb{R}^{n_z}$ into K clusters, RQ with depth D partitions the vector space into K^D clusters at most. That is, RQ with D has the same partition capacity as VQ with K^D codes. Thus, we can increase D for RQ to replace VQ with an exponentially growing codebook.

3.1.3 RQ-VAE

In Figure 2, we propose *RQ-VAE* to precisely quantize a feature map of an image. RQ-VAE is also comprised of the encoder-decoder architecture of VQ-VAE, but the VQ module is replaced with the RQ module above. Specifically, RQ-VAE with depth D represents a feature map $\mathbf{Z}$ as a stacked map of codes $\mathbf{M} \in [K]^{H \times W \times D}$ and extracts $\hat{\mathbf{Z}}^{(d)} \in \mathbb{R}^{H \times W \times n_z}$, which is the quantized feature map at depth d for $d \in [D]$ such that

$$\begin{aligned} \mathbf{M}_{hw} &= \mathcal{RQ}(E(\mathbf{X})_{hw}; \mathcal{C}, D), \\ \hat{\mathbf{Z}}^{(d)}_{hw} &= \sum_{d'=1}^{d} \mathbf{e}(\mathbf{M}_{hwd'}). \end{aligned} \tag{5}$$

For brevity, the quantized feature map $\hat{\mathbf{Z}}^{(D)}$ at depth D is also denoted by $\hat{\mathbf{Z}}$. Finally, the decoder G reconstructs the input image from $\hat{\mathbf{Z}}$ as $\hat{\mathbf{X}} = G(\hat{\mathbf{Z}})$.

Our RQ-VAE can make AR models to effectively generate high-resolution images with low computational costs. For a fixed downsampling factor f, RQ-VAE can produce more realistic reconstructions than VQ-VAE, since RQ-VAE can precisely approximate a feature map using a given codebook size. Note that the fidelity of reconstructed images is critical for the maximum quality of generated images. In addition, the precise approximation by RQ-VAE allows more increase of f and decrease of (H, W) than VQ-VAE, while preserving the reconstruction quality. Consequently, RQ-VAE enables an AR model to reduce its computational costs, increase the speed of image generation, and learn the long-range interactions of codes better.

Training of RQ-VAE To train the encoder E and the decoder G of RQ-VAE, we use the gradient descent with respect to the loss $\mathcal{L} = \mathcal{L}_{\text{recon}} + \beta \mathcal{L}_{\text{commit}}$ with a multiplicative factor $\beta > 0$, The *reconstruction loss* $\mathcal{L}_{\text{recon}}$ and the *commitment loss* $\mathcal{L}_{\text{commit}}$ are defined as

$$\mathcal{L}_{\text{recon}} = \|\mathbf{X} - \hat{\mathbf{X}}\|_2^2, \tag{6}$$

$$\mathcal{L}_{\text{commit}} = \sum_{d=1}^{D} \left\| \mathbf{Z} - \text{sg}\left[\hat{\mathbf{Z}}^{(d)}\right] \right\|_2^2, \tag{7}$$

where $\text{sg}[\cdot]$ is the stop-gradient operator, and the straight-through estimator [40] is used for the backpropagation through the RQ module. Note that $\mathcal{L}_{\text{commit}}$ is the sum of quantization errors from every d, not a single term $\|\mathbf{Z} - \text{sg}[\hat{\mathbf{Z}}]\|_2^2$. It aims to make $\hat{\mathbf{Z}}^{(d)}$ sequentially decrease the quantization error of $\mathbf{Z}$ as d increases. Thus, RQ-VAE approximates the feature map in a coarse-to-fine manner and keeps the training stable. The codebook $\mathcal{C}$ is updated by the exponential moving average of the clustered features [40].

Adversarial Training of RQ-VAE RQ-VAE is also trained with adversarial learning to improve the perceptual quality of reconstructed images. The patch-based adversarial loss [20] and the perceptual loss [21] are used together as described in the previous study [12]. We include the details in the supplementary material.

3.2. Stage 2: RQ-Transformer

In this section, we propose RQ-Transformer in Figure 2 to autoregressively predict a code stack of RQ-VAE. After we formulate the AR modeling of codes extracted by RQ-VAE, we introduce how our RQ-Transformer efficiently learns the stacked map of discrete codes. We also propose the training techniques for RQ-Transformer to prevent the exposure bias [34] in the training of AR models.

3.2.1 AR Modeling for Codes with Depth D

After RQ-VAE extracts a code map $\mathbf{M} \in [K]^{H \times W \times D}$, the raster-scan order [30] rearranges the spatial indices of $\mathbf{M}$ to a 2D array of codes $\mathbf{S} \in [K]^{T \times D}$ where $T = HW$. That is, $\mathbf{S}_t$, which is a t-th row of $\mathbf{S}$, contains D codes as

$$\mathbf{S}_t = (\mathbf{S}_{t1}, \cdots, \mathbf{S}_{tD}) \in [K]^D \quad \text{for } t \in [T]. \tag{8}$$

Regarding $\mathbf{S}$ as discrete latent variables of an image, AR models learn $p(\mathbf{S})$ which is autoregressively factorized as

$$p(\mathbf{S}) = \prod_{t=1}^{T} \prod_{d=1}^{D} p(\mathbf{S}_{td} | \mathbf{S}_{<t,d}, \mathbf{S}_{t,<d}). \tag{9}$$

3.2.2 RQ-Transformer Architecture

A naïve approach can unfold $\mathbf{S}$ into a sequence of length TD using the raster-scan order and feed it to the conventional transformer [41]. However, it neither leverages the reduced length of T by RQ-VAE and nor reduces the computational costs. Thus, we propose RQ-Transformer to efficiently learn the codes extracted by RQ-VAE with depth D. As shown in Figure 2, RQ-Transformer consists of *spatial transformer* and *depth transformer*.

Spatial Transformer The spatial transformer is a stack of *masked* self-attention blocks to extract a context vector that summarizes the information in previous positions. For the input of the spatial transformer, we reuse the learned codebook of RQ-VAE with depth D. Specifically, we define the input $\mathbf{u}_t$ of the spatial transformer as

$$\mathbf{u}_t = \text{PE}_T(t) + \sum_{d=1}^{D} \mathbf{e}(\mathbf{S}_{t-1,d}) \quad \text{for } t > 1, \tag{10}$$

where $\text{PE}_T(t)$ is a positional embedding for spatial position t. Note that the second term is equal to the quantized

feature vector of an image in Eq. 5. For the input at the first position, we define $\mathbf{u}_1$ as a learnable embedding, which is regarded as the start of a sequence. After the sequence $(\mathbf{u}_t)_{t=1}^T$ is processed by the spatial transformer, a context vector $\mathbf{h}_t$ encodes all information of $\mathbf{S}_{<t}$ as

$$\mathbf{h}_t = \text{SpatialTransformer}(\mathbf{u}_1, \cdots, \mathbf{u}_t). \tag{11}$$

Depth Transformer Given the context vector $\mathbf{h}_t$, the depth transformer autoregressively predicts D codes $(\mathbf{S}_{t1}, \cdots, \mathbf{S}_{tD})$ at position t. At position t and depth d, the input $\mathbf{v}_{td}$ of the depth transformer is defined as the sum of the code embeddings of up to depth $d-1$ such that

$$\mathbf{v}_{td} = \text{PE}_D(d) + \sum_{d'=1}^{d-1} \mathbf{e}(\mathbf{S}_{td'}) \quad \text{for } d > 1, \tag{12}$$

where $\text{PE}_D(d)$ is a positional embedding for depth d and shared at every position t. We do not use $PE_T(t)$ in $\mathbf{v}_{td}$, since the positional information is already encoded in $\mathbf{u}_t$. For $d = 1$, we use $\mathbf{v}_{t1} = \text{PE}_D(1) + \mathbf{h}_t$. Note that the second term in Eq. 12 corresponds to a quantized feature vector $\hat{\mathbf{Z}}_{hw}^{(d-1)}$ at depth $d-1$ in Eq. 5. Thus, the depth transformer predicts the next code for a finer estimation of $\hat{\mathbf{Z}}_t$ based on the previous estimations up to $d-1$. Finally, the depth transformer predicts the conditional distribution $\mathbf{p}_{td}(k) = p(\mathbf{S}_{td} = k | \mathbf{S}_{<t,d}, \mathbf{S}_{t,<d})$ as

$$\mathbf{p}_{td} = \text{DepthTransformer}(\mathbf{v}_{t1}, \cdots, \mathbf{v}_{td}). \tag{13}$$

RQ-Transformer is trained to minimize $\mathcal{L}_{AR}$, which is the negative log-likelihood (NLL) loss:

$$\mathcal{L}_{AR} = \mathbb{E}_{\mathbf{S}} \mathbb{E}_{t,d} \left[-\log p(\mathbf{S}_{td} | \mathbf{S}_{<t,d}, \mathbf{S}_{t,<d}) \right]. \tag{14}$$

Computational Complexity Our RQ-Transformer can efficiently learn and predict the $T \times D$ code maps of RQ-VAE, since RQ-Transformer has lower computational complexity than the naïve approach, which uses the unfolded 1D sequence of TD codes. When computing TD length of sequences, a transformer with N layers has $O(NT^2D^2)$ of computational complexity [41]. On the other hand, let us consider a RQ-Transformer with total N layers, where the number of layers in the spatial transformer and depth transformer is N_{spatial} and N_{depth}, respectively. Then, the spatial transformer requires $O(N_{\text{spatial}}T^2)$ and the depth transformer requires $O(N_{\text{depth}}TD^2)$, since the maximum sequence lengths for the spatial transformer and depth transformer are T and D, respectively. Hence, the computational complexity of RQ-Transformer is $O(N_{\text{spatial}}T^2 + N_{\text{depth}}TD^2)$, which is much less than $O(NT^2D^2)$. In Section 4.3, we show that our RQ-Transformer has a faster speed of image generation than previous AR models.

3.2.3 Soft Labeling and Stochastic Sampling

The exposure bias [34] is known to deteriorate the performance of an AR model due to the error accumulation from the discrepancy of predictions in training and inference. During inference, the prediction errors can also accumulate along with the depth D, since finer estimation of the feature vector becomes harder as d increases.

Thus, we propose *soft labeling* and *stochastic sampling* of codes from RQ-VAE to resolve the exposure bias. Scheduled sampling [2] is a way to reduce the discrepancy. However, it is unsuitable for a large-scale AR model, since multiple inferences are required at each training step and increase the training cost. Instead, we leverage the geometric relationship of code embeddings in RQ-VAE. We define a categorical distribution on $[K]$ conditioned by a vector $\mathbf{z} \in \mathbb{R}^{n_z}$ as $\mathcal{Q}_\tau(k|\mathbf{z})$, where $\tau > 0$ is a temperature

$$\mathcal{Q}_\tau(k|\mathbf{z}) \propto e^{-\|\mathbf{z} - \mathbf{e}(k)\|_2^2 / \tau} \quad \text{for } k \in [K]. \tag{15}$$

As τ approaches zero, $\mathcal{Q}_\tau$ is sharpened and converges to the one-hot distribution $\mathcal{Q}_0(k|\mathbf{z}) = \mathbf{1}[k = \mathcal{Q}(\mathbf{z}; \mathcal{C})]$.

Soft Labeling of Target Codes Based on the distance between code embeddings, soft labeling is used to improve the training of RQ-Transformer by explicit supervision on the geometric relationship between the codes in RQ-VAE. For a position t and a depth d, let $\mathbf{Z}_t$ be a feature vector of an image and $\mathbf{r}_{t,d-1}$ be a residual vector at depth $d-1$ in Eq. 4. Then, the NLL loss in Eq. 14 uses the one-hot label $\mathcal{Q}_0(\cdot|\mathbf{r}_{t,d-1})$ as the supervision of $\mathbf{S}_{td}$. Instead of the one-hot labels, we use the softened distribution $\mathcal{Q}_\tau(\cdot|\mathbf{r}_{t,d-1})$.

Stochastic Sampling for Codes of RQ-VAE Along with the soft labeling above, we propose stochastic sampling of the code map from RQ-VAE to reduce the discrepancy in training and inference. Instead of the deterministic code selection of RQ in Eq. 4, we select the code $\mathbf{S}_{td}$ by sampling from $\mathcal{Q}_\tau(\cdot|\mathbf{r}_{t,d-1})$. Note that our stochastic sampling is equivalent to the original code selection of RQ in the limit of $\tau \to 0$. The stochastic sampling provides different compositions of codes $\mathbf{S}$ for a given feature map of an image.

4. Experiments

In this section, we empirically validate our model for high-quality image generation. We evaluate our model on unconditional image generation benchmarks in Section 4.1 and conditional image generation in Section 4.2. The computational efficiency of RQ-Transformer is shown in Section 4.3. We also conduct an ablation study to understand the effectiveness of RQ-VAE in Section 4.4.

For a fair comparison, we adopt the model architecture of VQ-GAN [12]. However, since RQ-VAEs convert

Figure 3. Generated images by our models. First row: LSUN-{cat, bedroom, church}. Second row: FFHQ, ImageNet and CC-3M. The text conditions of CC-3M are "Mountains and hills reflecting over a surface," and "Businessman with a paper bag on head," respectively.

Table 1. Comparison of FIDs for unconditional image generation on LSUN-{Cat, Bedroom, Church} [43] and FFHQ [23].

	Cat	Bedroom	Church	FFHQ
DDPM [19]	19.75	4.90	7.89	-
ImageBART [11]	15.09	5.51	7.32	9.57
StyleGAN2 [24]	7.25	2.35	3.86	3.8
DCT [29]	-	6.40	7.56	13.06
VQ-GAN [12]	17.31	6.35	7.81	11.4
RQ-Transformer	8.64	3.04	7.45	10.38

256×256×3 RGB images into 8×8×4 codes, we add an encoder and decode block to RQ-VAE and further decreases the resolution of the feature map by half. We include other details of implementation in the supplementary material.

4.1. Unconditional Image Generation

The results of unconditional image generation is evaluated on the LSUN-{cat, bedroom, church} [43] and FFHQ [23] datasets. The codebook size K is 2048 for FFHQ and 16384 for LSUN. For FFHQ, RQ-VAE is trained from scratch during 100 epochs, and early stopping is used for RQ-Transformer when the validation loss is minimized, since the small size of FFHQ leads to overfitting of AR models. For the LSUN datasets, we use a pretrained RQ-VAE on ImageNet and finetune the model for one epoch on each dataset. Considering the dataset size, we use RQ-Transformer of 612M parameters for LSUN-{cat, bedroom} and 370M parameters for LSUN-church and FFHQ. For the evaluation measure, we use Frechet Inception Distance (FID) [18] between 50K generated samples and all training samples. We also use top-k and top-p sampling to report the best performance [11, 12].

Table 1 shows that our model outperforms the other AR models on unconditional image generation. For small-scale datasets such as LSUN-church and FFHQ, our model outperforms DCT [29] and VQ-GAN [12] with marginal improvements. However, for a larger scale of datasets such as LSUN-{cat, bedroom}, our model significantly outperforms other AR models and diffusion-based models [11, 19]. We conjecture that the performance improvement comes from the shorter sequence length by RQ-VAE, since SQ-Transformer can easily learn the long-range interactions between codes in the short length of the sequence. In the first two rows of Figure 3, we show that RQ-Transformer can unconditionally generate high-quality images.

4.2. Conditional Image Generation

We use ImageNet [7] and CC-3M [39] for a class- and text-conditioned image generation, respectively. We train RQ-VAE with K=16,384 on ImageNet training data for 10 epochs and reuse the trained RQ-VAE for CC-3M. For ImageNet, we also use RQ-VAE trained for 50 epochs to examine the effect of improved reconstruction quality on image generation quality of RQ-Transformer in Table 2. For conditioning, we append the embeddings of class and text conditions to the start of input for spatial transformer. The texts of CC-3M are represented as a sequence of at most 32 tokens using a byte pair encoding [37, 42].

Table 2 shows that our model significantly outperforms previous models on ImageNet. Our RQ-Transformer of 480M parameters outperforms the previous AR models including VQ-VAE2 [35], DCT [29], and VQ-GAN [12] without rejection sampling, although our model has 3× less parameters than VQ-GAN. Our stochastic sampling is also effective for performance improvement, while RQ-Transformer without it still outperforms other AR models. RQ-Transformer of 1.4B parameters achieves 11.56 of FID score without rejection sampling. When we improve the reconstruction quality of RQ-VAE by increasing its training

Table 2. Comparison of FIDs and ISs for class-conditioned image generation on ImageNet [7]. † denotes a model without our stochastic sampling and soft labeling. ‡ denotes the use of rejection sampling or gradient guidance by pretrained classifier. $*$ denotes the use of RQ-VAE trained for 50 epochs.

	Params	FID	IS
without rejection sampling or gradient guidance			
ADM [9]	554M	10.94	101.0
ImageBART [11]	3.5B	21.19	61.6
BigGAN [3]	164M	7.53	168.6
BigGAN-deep [3]	112M	6.84	203.6
VQ-VAE2 [35]	13.5B	∼31	∼45
DCT [29]	738M	36.5	n/a
VQ-GAN [12]	1.4B	15.78	74.3
RQ-Transformer	480M	15.72	86.8±1.4
RQ-Transformer†	821M	14.06	95.8±2.1
RQ-Transformer	821M	13.11	104.3±1.5
RQ-Transformer	1.4B	11.56	112.4±1.1
RQ-Transformer*	1.4B	8.71	119.0±2.5
RQ-Transformer*	3.8B	7.55	134.0±3.0
with rejection sampling or gradient guidance			
ADM‡ [9]	554M	4.59	186.7
ImageBART‡ [11]	3.5B	7.44	273.5±4.1
VQ-GAN‡ [12]	1.4B	5.20	280.3±5.5
RQ-Transformer‡	1.4B	4.45	326.0±3.5
RQ-Transformer*‡	1.4B	3.89	337.5±4.6
RQ-Transformer*‡	3.8B	3.80	323.7±2.8

Table 3. Comparison of FID and CLIP score [32] on the validation data of CC-3M [39] for text-conditioned image generation.

	Params	FID	CLIP-s
VQ-GAN [12]	600M	28.86	0.20
ImageBART [11]	2.8B	22.61	0.23
RQ-Transformer	654M	12.33	0.26

Table 4. Comparison of FIDs between ImageNet validation images and their reconstructed images according to codebook size (K) and the shape of code map $H \times W \times D$. † denotes the reproduced performance, and $*$ denotes 50 epochs of training.

	$H \times W \times D$	K	rFID
VQ-GAN [12]	16×16×1	16,384	4.90
VQ-GAN†	16×16×1	16,384	4.32
VQ-GAN	8×8×1	16,384	17.95
VQ-GAN	8×8×1	65,536	17.66
VQ-GAN	8×8×1	131,072	17.09
RQ-VAE	8×8×2	16,384	10.77
RQ-VAE	8×8×4	16,384	4.73
RQ-VAE*	8×8×4	16,384	3.20
RQ-VAE	8×8×8	16,384	2.69

epoch from 10 to 50, RQ-Transformer of 1.4B parameters achieves 8.71 of FID. Moreover, when we further increase the model size to 3.8B, RQ-Transformer achieves 7.55 of FID score without rejection sampling and is competitive with BigGAN [3]. When ResNet-101 [16] is used for rejection sampling with 5% and 12.5% of acceptance rates for 1.4B and 3.8B parameters, respectively, our model outperforms ADM [9] and achieves the state-of-the-art FID score.

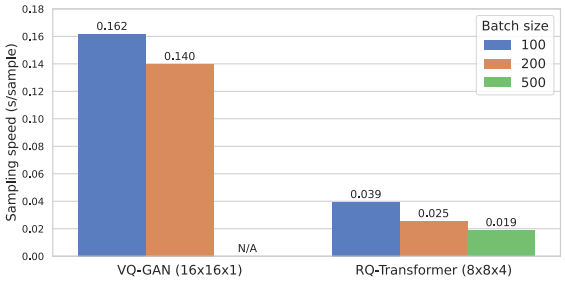

Figure 4. The sampling speed of RQ-Transformer with 1.4B parameters according to batch size and code map shape.

RQ-Transformer can also generate high-quality images based on various text conditions of CC-3M. RQ-Transformer shows significantly higher performance than VQ-GAN with a similar number of parameters. In addition, although RQ-Transformer has 23% of parameters, our model significantly outperforms ImageBART [11] on both FID and CLIP score [32] (with ViT-B/32 [10]). Figure 3 shows that RQ-Transformer trained on CC-3M can generate high-quality images using various text conditions. In addition, the text conditions in Figure 1 are novel compositions of visual concepts, which are unseen in training.

4.3. Computational Efficiency of RQ-Transformer

In Figure 4, we evaluate the sampling speed of RQ-Transformer to compare with VQ-GAN. Both the models have 1.4B parameters. The shape of the input code map for VQ-GAN and RQ-Transformer are set to be 16×16×1 and 8×8×4, respectively. We use a single NVIDIA A100 GPU for each model to generate 5000 samples with 100, 200, and 500 of batch size. The reported speeds in Figure 4 do not include the decoding time of the stage 1 model to focus on the effect of RQ-Transformer architecture. The decoding time of VQ-GAN and RQ-VAE is about 0.008 sec/image.

For the batch size of 100 and 200, RQ-Transformer shows 4.1× and 5.6× speed-up compared with VQ-GAN. Moreover, thanks to the memory saving from the short sequence length of RQ-VAE, RQ-Transformer can increase the batch size into 500, which is not allowed for VQ-GAN. RQ-Transformer can further accelerate the sampling speed up to 0.02 seconds per image, which is 7.3× faster than VQ-GAN with batch size 200. Thus, RQ-Transformer is more computationally efficient than previous AR models, while achieving state-of-the-art results of image generation.

Figure 5. The examples of coarse-to-fine approximation by RQ-VAE. The first example is the original image, and the others are reconstructed from $\hat{\mathbf{Z}}^{(d)}$. As d increases, the reconstructed images become clear and include fine-grained details of the original image.

4.4. Ablation Study on RQ-VAE

We conduct an ablation study to understand the effect of RQ with respect to the codebook size (K) and the shape of the code map ($H \times W \times D$). We measure the rFID, which is FID between original images and reconstructed images, on ImageNet validation data. Table 4 shows that increasing the quantization depth D is more effective to improve the reconstruction quality than increasing the codebook size K. Here, we remark that RQ-VAE with $D=1$ is equivalent to VQ-GAN. For a fixed codebook size K=16,384, the rFID significantly deteriorates as the spatial resolution $H \times W$ is reduced from 16×16 to 8×8. Even when the codebook size is increased to K=131,072, the rFID cannot recover the rFID with 16×16 feature maps, since the restoration of rFID requires the codebook of size K=16,384^4 in terms of the rate-distortion trade-off. Contrastively, note that the rFIDs are significantly improved when we increase the quantization depth D with a codebook of fixed size K=16,384. Thus, our RQ-VAE can further reduce the spatial resolution than VQ-GAN, while conserving the reconstruction quality. Although RQ-VAE with $D > 4$ can further improve the reconstruction quality, we use RQ-VAE with 8×8×4 code map for AR modeling of images, considering the computational costs of RQ-Transformer. In addition, the longer training of RQ-VAE can further improve the reconstruction quality, but we train RQ-VAE for 10 epochs as the default due to its increased training time.

Figure 5 substantiates our claim that RQ-VAE conducts the coarse-to-fine estimation of feature maps. For example, Figure 5 shows the reconstructed images $G(\hat{\mathbf{Z}}^{(d)})$ of a quantized feature map at depth d in Eq. 4. When we only use the codes at $d = 1$, the reconstructed image is blurry and only contains coarse information of the original image. However, as d increases and the information of remaining codes is sequentially added, the reconstructed image includes more clear and fine-grained details.

5. Conclusion

Discrete representation of visual images is important for an AR model to generate high-resolution images. In this work, we have proposed RQ-VAE and RQ-Transformer for high-quality image generation. Under a fixed codebook size, RQ-VAE can precisely approximate a feature map of an image to represent the image as a short sequence of codes. Thus, RQ-Transformer effectively learns to predict the codes to generate high-quality images with low computational costs. Consequently, our approach outperforms the previous AR models on various image generation benchmarks such as LSUNs, FFHQ, ImageNet, and CC-3M.

Our study has three main limitations. First, our model does not outperform StyleGAN2 [24] on unconditional image generation, especially with a small-scale dataset such as FFHQ, due to overfitting of AR models. Thus, regularizing AR models is worth exploration for high-resolution image generation on a small dataset. Second, our study does not enlarge the model and training data for text-to-image generation. As a previous study [17, 33] shows that a huge transformer can effectively learn the zero-shot text-to-image generation, increasing the number of parameters is an interesting future work. Third, AR models can only capture unidirectional contexts to generate images compared to other generative models. Thus, modeling of bidirectional contexts can further improve the quality of image generation and enable AR models to be used for image manipulation such as image inpainting and outpainting [11].

Although our study significantly reduces the computational costs for AR modeling of images, training of large-scale AR models is still expensive, consumes high amounts of electrical energy, and can leave a huge carbon footprint, as the scale of model and training dataset becomes large. Thus, efficient training of large-scale AR models is still worth exploration to avoid environmental pollution.

6. Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2018-0-01398: Development of a Conversational, Self-tuning DBMS; No.2021-0-00537: Visual Common Sense).

References

[1] Artem Babenko and Victor Lempitsky. Additive quantization for extreme vector compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 931–938, 2014. 2

[2] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *arXiv preprint arXiv:1506.03099*, 2015. 5

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. 7

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. 2

[5] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning*, pages 1691–1703. PMLR, 2020. 1, 2

[6] Gregory F Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial intelligence*, 42(2-3):393–405, 1990. 2

[7] 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. 6, 7

[8] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020. 2, 3

[9] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *arXiv preprint arXiv:2105.05233*, 2021. 7

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 7

[11] Patrick Esser, Robin Rombach, Andreas Blattmann, and Björn Ommer. Imagebart: Bidirectional context with multinomial diffusion for autoregressive image synthesis, 2021. 1, 2, 6, 7, 8

[12] 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, 4, 5, 6, 7

[13] Sohrab Ferdowsi, Slava Voloshynovskiy, and Dimche Kostadinov. Regularized residual quantization: a multi-layer sparse dictionary learning approach. *arXiv preprint arXiv:1705.00522*, 2017. 2

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 2

[15] R. Gray. Vector quantization. *IEEE ASSP Magazine*, 1(2):4–29, 1984. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[17] Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020. 8

[18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 6

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020. 6

[20] 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*, pages 1125–1134, 2017. 2, 4

[21] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 4

[22] Biing-Hwang Juang and A Gray. Multiple stage vector quantization for speech coding. In *ICASSP'82. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 7, pages 597–600. IEEE, 1982. 2

[23] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 6

[24] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 6, 8

[25] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017. 2

[26] Yue Li, Wenrui Ding, Chunlei Liu, Baochang Zhang, and Guodong Guo. Trq: Ternary neural networks with residual

quantization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8538–8546, 2021. 2

[27] Zefan Li, Bingbing Ni, Wenjun Zhang, Xiaokang Yang, and Wen Gao. Performance guaranteed network acceleration via high-order residual quantization. In *Proceedings of the IEEE international conference on computer vision*, pages 2584–2592, 2017. 2

[28] Julieta Martinez, Holger H Hoos, and James J Little. Stacked quantizers for compositional vector compression. *arXiv preprint arXiv:1411.2173*, 2014. 2

[29] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter Battaglia. Generating images with sparse representations. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7958–7968. PMLR, 18–24 Jul 2021. 2, 6, 7

[30] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *Proceedings of The 33rd International Conference on Machine Learning*, 2016. 1, 2, 4

[31] Aäron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 4797–4805, 2016. 2

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. 7

[33] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8821–8831. PMLR, 18–24 Jul 2021. 1, 2, 8

[34] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. 2, 4, 5

[35] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Advances in neural information processing systems*, pages 14866–14876, 2019. 2, 6, 7

[36] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. Pixelcnn++: A pixelcnn implementation with discretized logistic mixture likelihood and other modifications. In *ICLR*, 2017. 2

[37] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, 2016. 6

[38] Claude E Shannon et al. Coding theorems for a discrete source with a fidelity criterion. *IRE Nat. Conv. Rec*, 4(142-163):1, 1959. 1, 3

[39] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia, July 2018. Association for Computational Linguistics. 6, 7

[40] Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural discrete representation learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 1, 2, 4

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 4, 5

[42] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, Oct. 2020. Association for Computational Linguistics. 6

[43] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 6