Draft-and-Revise: Effective Image Generation with Contextual RQ-Transformer

Doyup Lee[*]
Kakao Brain
doyup.lee@kakaobrain.com

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@dblab.postech.ac.kr

Abstract

Although autoregressive models have achieved promising results on image generation, their unidirectional generation process prevents the resultant images from fully reflecting global contexts. To address the issue, we propose an effective image generation framework of *Draft-and-Revise* with *Contextual RQ-transformer* to consider global contexts during the generation process. As a generalized VQ-VAE, RQ-VAE first represents a high-resolution image as a sequence of discrete code stacks. After code stacks in the sequence are randomly masked, Contextual RQ-Transformer is trained to infill the masked code stacks based on the unmasked contexts of the image. Then, we propose the two-phase decoding, Draft-and-Revise, for Contextual RQ-Transformer to generate an image, while fully exploiting the global contexts of the image during the generation process. Specifically. in the *draft* phase, our model first focuses on generating diverse images despite rather low quality. Then, in the *revise* phase, the model iteratively improves the quality of images, while preserving the global contexts of generated images. In experiments, our method achieves state-of-the-art results on conditional image generation. We also validate that the Draft-and-Revise decoding can achieve high performance by effectively controlling the quality-diversity trade-off in image generation.

1 Introduction

Learning discrete representations of images enables autoregressive (AR) models to achieve promising results on image generation. Here, an image is encoded into a feature map, which is represented as a sequence of discrete codes [13, 36] or code stacks [26]. Then, an AR model generates a sequence of codes in the raster scan order and decodes the codes into an image. Consequently, AR models show high performance and scalability on large-scale datasets [13, 26, 29].

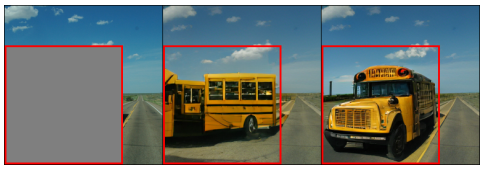


Figure 1: Examples of image inpainting by an AR model (middle) and ours (right).

Despite the promising results of AR models, we postulate that the ability of AR models is limited due to the lack of considering global contexts in the generation process. Specifically, since AR models generate images by sequentially predicting the

[*]Equal contribution
[†]Corresponding author

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

next code and attending to only precedent codes generated, they neither exploit the later part of the generated image nor consider the global contexts during generation. For example, Figure 1 (middle) shows that an AR model fails to generate a coherent image, when it is asked to inpaint the masked region of Figure 1 (left) with a school bus. Such a failure is due to the inability of AR models to refer to the context of traffic lane on the right side of the masked region.

To address this issue, we propose an effective image generation framework, *Draft-and-Revise*, with a contextual transformer to exploit the global contexts of images. Given a randomly masked image, the contextual transformer is first trained to infill the masks by bidirectional self-attentions similarly to BERT [8]. To fully leverage the contextual prediction in generation, we propose Draft-and-Revise decoding which has two phases, *draft* and *revise*, imitating the image generation process of a human expert who draws a draft first and iteratively revises the draft to improve its quality. In the draft phase, the model first infills an empty image to generate a draft image with diverse contents despite the rather low-quality. In the revise phase, the visual quality of the draft is iteratively improved, while the global contexts of the draft are preserved and exploited. Consequently, our *Draft-and-Revise* with contextual transformer effectively generates high-quality images with diverse contents.

We use residual-quantized VAE (RQ-VAE) [26] to implement our image generation framework, since RQ-VAE generalizes vector-quantized VAE (VQ-VAE) [36] by representing an image as a sequence of code stacks instead of a sequence of codes. Then, we propose Contextual RQ-Transformer as a contextual transformer for masked code stack modeling of RQ-VAE. Specifically, given a sequence of randomly masked code stacks, Contextual RQ-Transformer first uses a bidirectional transformer to capture the global contexts of unmasked code stacks. Based on the global contexts, the masked code stacks are predicted in parallel, while the codes in each masked code stack are sequentially predicted. In experiments, our Draft-and-Revise framework with Contextual RQ-Transformer achieves state-of-the-art results on conditional image generation and remarkable improvements on image inpainting. In addition, we demonstrate that Draft-and-Revise decoding can effectively control the quality-diversity trade-off in image generation to achieve high performance.

The main contributions of this paper are summarized as follows. 1) We propose an intuitive and powerful framework, *Contextual RQ-Transformer*, for masked code stack modeling of RQ-VAE based on a bidirectional transformer. 2) We propose a novel two-phase decoding, *Draft-and-Revise*, for bidirectional transformers to fully exploit the global contexts during image generation and achieve state-of-the-art results on class- and text-conditional image generation benchmarks. 3) An extensive ablation study validates the effectiveness of Draft-and-Revise decoding on controlling the quality-diversity trade-off and its capability to generate high-quality images with diverse contents.

2 Related Work

Discrete Representation for Image Generation By representing an image as a sequence of codes, VQ-VAE [36] becomes an important part for high-resolution image generation [6, 10, 15, 26, 29, 36], but suffers from low quality of reconstructed images. However, VQGAN [13] significantly improves the perceptual quality of reconstructed images by adding the adversarial and perceptual losses into the training objective of VQ-VAE. As a generalized approach of VQ-VAE and VQGAN, RQ-VAE [26] represents an image as a sequence of code stacks, which consists of ordered codes, and reduces the sequence length, while preserving the reconstruction quality. Then, RQ-Transformer [26] achieves high performance with lower computational costs on generating high-resolution images. However, as an AR model of RQ-VAE, RQ-Transformer cannot capture the global contexts of generated images.

Generation Tasks with Bidirectional Transformers To overcome the limitation of AR models on unidirectional architecture, bidirectional transformers have been studied for generative tasks. After a bidirectional transformer is trained to infill a random mask as the masked token modeling of BERT [8], an effective decoding method has been proposed for bidirectional transformers to generate texts [14, 33], images [6, 38], or videos [16]. For image generation, M6-UFC [38] first incorporates the masked token modeling in the codes of VQGAN [13] for image generation. MaskGIT [6] proposes the confidence-based decoding to achieve high performance on ImageNet [7]. Similar to M6-UFC [38] and MaskGIT [6], we also incorporate the masked token modeling of RQ-VAE in a bidirectional transformer for image generation. However, instead of the confidence-based decoding [6, 38], we propose a novel two-phase decoding for a bidirectional transformer to achieve state-of-the-art performance and validate its effectiveness. Recently, discrete diffusion models [1, 4, 12, 15] also uses

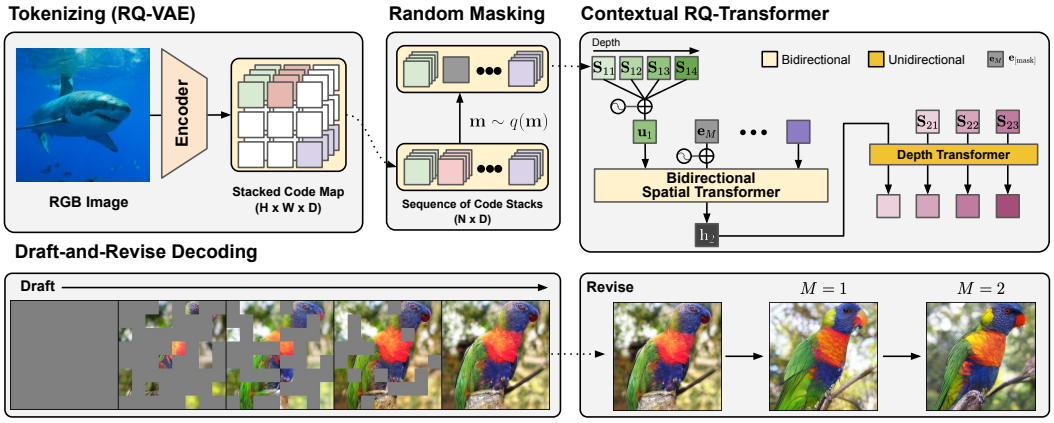


Figure 2: The overview of Draft-and-Revise framework with Contextual RQ-Transformer. Our framework exploits global contexts of images to generate high-quality images with diverse contents.

bidirectional transformers to generate an image, and the training with an absorbing state [1] is also the same to infill randomly masked sequence [4, 6]. However, different from the reverse process of diffusion models, our decoding method has explicit two phases to generate high-quality images with diverse contents.

3 *Draft-and-Revise* Framework for Effective Image Generation

In this section, we propose our *Draft-and-Revise* framework for effective image generation using bidirectional contexts of images. We first review RQ-VAE [26] as a generalization of VQ-VAE. Then, we propose Contextual RQ-Transformer which is trained to infill a randomly masked sequence of code stacks of RQ-VAE by understanding bidirectional contexts of unmasked parts in the sequence. Lastly, we propose *draft-and-revise* decoding for a bidirectional transformer to effectively generate high-quality images exploiting global contexts of images. Figure 2 provides the overview of our proposed framework, including Contextual RQ-Transformer and Draft-and-Revise decoding.

3.1 Residual-Quantized Variational Autoencoder (RQ-VAE)

RQ-VAE [26] represents an image as a sequence of code stacks. Let a codebook $\mathcal{C} = \{(k, \mathbf{e}(k))\}_{k \in [K]}$ include pairs of a code k and its code embedding $\mathbf{e}(k) \in \mathbb{R}^{n_z}$, where $K = |\mathcal{C}|$ is the codebook size and n_z is the dimensionality of $\mathbf{e}(k)$. Given a vector $\mathbf{z} \in \mathbb{R}^{n_z}$, $\mathcal{Q}(\mathbf{z}; \mathcal{C})$ is defined as the code of $\mathbf{z}$:

$$\mathcal{Q}(\mathbf{z}; \mathcal{C}) = \arg\min_{k} \|\mathbf{z} - \mathbf{e}(k)\|_2^2 \,. \tag{1}$$

Then, RQ with depth D represents a vector as a *code stack* which consists of D codes:

$$\mathcal{RQ}(\mathbf{z}; \mathcal{C}, D) = (k_1, \cdots, k_D) \in [K]^D, \tag{2}$$

where k_d is the d-th code of $\mathbf{z}$. Specifically, RQ first initializes the 0-th residual vector as $\mathbf{r}_0 = \mathbf{z}$, and then recursively discretizes a residual vector $\mathbf{r}_{d-1}$ and computes the next residual vector $\mathbf{r}_d$ as

$$k_d = \mathcal{Q}(\mathbf{r}_{d-1}; \mathcal{C}), \qquad \mathbf{r}_d = \mathbf{r}_{d-1} - \mathbf{e}(k_d), \tag{3}$$

for $d \in [D]$. Finally, $\mathbf{z}$ is approximated by the sum of the D code embeddings $\hat{\mathbf{z}} := \sum_{d=1}^{D} \mathbf{e}(k_d)$. We remark that RQ is a generalized version of VQ, as RQ with $D = 1$ is equivalent to VQ. For $D > 1$, RQ conducts a finer approximation of $\mathbf{z}$ as the quantization errors are sequentially reduced as d increases. Here, the coarse-to-fine approximation ensures the D codes to be sequentially dependent.

RQ-VAE represents an image as a map of code stacks. Specifically, a given image $\mathbf{X}$ is first converted to a low-resolution feature map $\mathbf{Z} = E(\mathbf{X}) \in \mathbb{R}^{H \times W \times n_z}$, and then each feature vector $\mathbf{Z}_{hw}$ at spatial position (h, w) is discretized into a code stack by RQ with depth D. As a result, we get a map of code stacks $\mathbf{S} \in [K]^{H \times W \times D}$. Further details of RQ-VAE are referred to Appendix.

3.2 Contextual Transformer for Image Generation with Global Contexts

As a bidirectional transformer for RQ-VAE, we propose Contextual RQ-Transformer for image generation based on a contextual understanding of images. First, we adopt the pretraining of BERT [8] to formulate a masked code stack modeling of RQ-VAE. Then, we introduce how Contextual RQ-Transformer infills the randomly masked code stacks after reading the given contextual information.

3.2.1 Masked Code Stack Modeling of RQ-VAE

By adopting the pretraining of BERT [8], we formulate the masked code stack modeling of RQ-VAE with a contextual transformer to generate an image by iterative mask-infilling as non-AR models [14]. We first convert the map $\mathbf{S} \in [K]^{H \times W \times D}$ into a sequence of code stacks $\mathbf{S}' \in [K]^{N \times D}$ using the raster-scan ordering, where $N = HW$ and $\mathbf{S}'_n = (\mathbf{S}'_{n1}, \cdots, \mathbf{S}'_{nD}) \in [K]^D$ for $n \in [N]$. We denote $\mathbf{S}'$ as $\mathbf{S}$ for the brevity of notation. A mask vector $\mathbf{m}$ is defined as a binary vector $\mathbf{m} \in \{0,1\}^N$ to indicate the spatial positions to be masked. Then, the masked sequence $\mathbf{S}_{\setminus \mathbf{m}}$ of $\mathbf{S}$ by $\mathbf{m}$ is defined as

$$(\mathbf{S}_{\setminus \mathbf{m}})_n = \begin{cases} \mathbf{S}_n & \text{if } \mathbf{m}_n = 0 \\ [\text{MASK}]^D & \text{if } \mathbf{m}_n = 1 \end{cases}, \tag{4}$$

where [MASK] is a mask token to substitute for $\mathbf{S}_{nd}$ if $\mathbf{m}_n = 1$. Given a random mask vector $\mathbf{m} \sim q(\mathbf{m})$, the masked code stacks given $\mathbf{S}_{\setminus \mathbf{m}}$ are modeled as

$$\prod_{n:\mathbf{m}_n=1} p(\mathbf{S}_n | \mathbf{S}_{\setminus \mathbf{m}}) = \prod_{n:\mathbf{m}_n=1} \prod_{d=1}^{D} p(\mathbf{S}_{nd} | \mathbf{S}_{n,<d}, \mathbf{S}_{\setminus \mathbf{m}}), \tag{5}$$

where $q(\mathbf{m})$ is a mask distribution where the masking portion $\sum_{n=1}^{N} \mathbf{m}_i / N$ in $(0,1]$ as well as the masking positions are randomly chosen. Instead of fixing the portion to 15% as in BERT, training a model with a random masking portion from $(0,1]$ enables the model to generate new images based on various masking patterns including $\mathbf{m}_n = 1$ for all n. We explain the details of $q(\mathbf{m})$ in Section 3.2.3.

The left-hand side of Eq. 5 implies that all masked code stacks can be decoded in parallel based on the contexts of $\mathbf{S}_{\setminus \mathbf{m}}$. If $D = 1$, Eq. 5 becomes equivalent to conventional masked token modeling of texts [8] and images [6, 16] where a single token at each masked position is predicted. For $D > 1$, the D codes of $\mathbf{S}_n$ are autoregressively predicted, as they are sequentially computed in Eq. 3 for a coarse-to-fine approximation and hence well-suited for an AR prediction. We show that the effectiveness of our framework is generalizable regardless of D in Appendix B. In addition, the parallel decoding can control the trade-off between the quality and sampling speed of image generation.

3.2.2 Contextual RQ-Transformer

We modify the previous RQ-Transformer [26] for masked code stack modeling with bidirectional contexts in Eq. 5. Contextual RQ-Transformer consists of *Bidirectional Spatial Transformer* and *Depth Transformer*: Bidirectional Spatial Transformer understands contextual information in the unmasked code stacks using bidirectional self-attentions, and Depth Transformer infills the masked code stacks in parallel, by autoregressively predicting the D codes at each position.

***Bidirectional* Spatial Transformer** Given a masked sequence of code stacks $\mathbf{S}_{\setminus \mathbf{m}}$, bidirectional spatial transformer first embeds the masked sequence $\mathbf{S}_{\setminus \mathbf{m}}$ using the code embeddings of RQ-VAE as

$$\mathbf{u}_n = \text{PE}_N(n) + \begin{cases} \sum_{d=1}^{D} \mathbf{e}(\mathbf{S}_{nd}) & \text{if } \mathbf{m}_n = 0 \\ \mathbf{e}_{[\text{MASK}]} & \text{if } \mathbf{m}_n = 1 \end{cases}, \tag{6}$$

where $\text{PE}_N(n)$ is an embedding for position n, and $\mathbf{e}_{[\text{MASK}]} \in \mathbb{R}^{n_z}$ is a mask embedding. Then, the bidirectional self-attention blocks, $f_\theta^{\text{spatial}}$, extracts the context vector $\mathbf{h}_n$ to predict a code stack $\mathbf{S}_n$,

$$(\mathbf{h}_1, \cdots, \mathbf{h}_N) = f_\theta^{\text{spatial}}(\mathbf{u}_1, \cdots, \mathbf{u}_N). \tag{7}$$

Depth Transformer The code stack at a masked position $\mathbf{S}_n = (\mathbf{S}_{n1}, \cdots, \mathbf{S}_{nD})$ is autoregressively predicted. The input of depth transformer $(\mathbf{v}_{nd})_{d=1}^{D}$ is defined as

$$\mathbf{v}_{nd} = \text{PE}_D(d) + \begin{cases} \mathbf{h}_n & \text{if } d = 1 \\ \sum_{d'=1}^{d-1} \mathbf{e}(\mathbf{S}_{nd'}) & \text{if } d > 1 \end{cases} \tag{8}$$

Algorithm 1 UPDATE of $\mathbf{S}$

Require: A sequence of code stacks $\mathbf{S}$, a partition $\mathbf{\Pi} = (\mathbf{m}^1, \cdots, \mathbf{m}^T)$, a model θ

for $t = 1, \cdots, T$ **do**
  Sample $\mathbf{S}_n \sim p_\theta(\mathbf{S}_n | \mathbf{S}_{\setminus \mathbf{m}^t}) \quad \forall n : \mathbf{m}^t_n = 1$ ▷ update the codes at masked positions
end for
return $\mathbf{S}$

Algorithm 2 Draft-and-Revise decoding

Require: Partition sampling distributions p_{draft} and p_{rev}, the number of revision iterations M

/* draft phase */
$\mathbf{S}^{\text{empty}} \leftarrow ([\text{MASK}], \cdots, [\text{MASK}])^N$ ▷ initialize empty code map
Sample $\mathbf{\Pi} \sim p(\mathbf{\Pi}; T_{\text{draft}})$
$\mathbf{S}^{\text{draft}} \leftarrow \text{UPDATE}(\mathbf{S}^{\text{empty}}, \mathbf{\Pi}; \theta)$ ▷ generate a draft code map
/* revision phase */
$\mathbf{S}^0 \leftarrow \mathbf{S}^{\text{draft}}$
for $m = 1, \cdots, M$ **do**
  Sample $\mathbf{\Pi} \sim p(\mathbf{\Pi}; T_{\text{revise}})$
  $\mathbf{S}^m \leftarrow \text{UPDATE}(\mathbf{S}^{m-1}, \mathbf{\Pi}; \theta)$ ▷ iteratively revise the code map
end for
return $\mathbf{S}^M$

where $\text{PE}_D(d)$ is the positional embedding for depth d. Then, depth transformer f_θ^{depth}, which consists of causal attention blocks, outputs the logit $\mathbf{p}_{nd}$ to predict the d-th code $\mathbf{S}_{nd}$ at position n as

$$\mathbf{p}_{nd} = f_\theta^{\text{depth}}(\mathbf{v}_{n1}, \cdots, \mathbf{v}_{nd}) \quad \text{and} \quad p_\theta(\mathbf{S}_{nd} = k | \mathbf{S}_{n,<d}, \mathbf{S}_{\setminus \mathbf{m}}) = \text{softmax}(\mathbf{p}_{nd})_k. \tag{9}$$

We remark that the architecture of Contextual RQ-Transformer subsumes bidirectional transformers. Specifically, RQ-Transformer with the depth $D = 1$ is equivalent to a bidirectional transformer since the depth transformer becomes a multilayer perceptron with layer normalization [2].

3.2.3 Training of Contextual RQ-Transformer

For the training of Contextual RQ-Transformer, let us define a mask distribution $q(\mathbf{m})$ with a mask scheduling function γ. Following previous approaches [6, 14, 16], the scheduling function γ is chosen to be decreasing and to satisfy $\gamma(0) = 1$ and $\gamma(1) = 0$. Then, a mask $\mathbf{m} \sim q(\mathbf{m})$ is specified as

$$r \sim \text{Unif}([0, 1)) \quad \text{and} \quad \mathbf{m} \sim \text{Unif}(\{\mathbf{m} : |\mathbf{m}| = \lceil \gamma(r) \cdot N \rceil\}), \tag{10}$$

where $|\mathbf{m}| = \sum_{n \in [N]} \mathbf{m}_n$ is the count of masked positions. Finally, the training objective of Contextual RQ-Transformer is to minimize the negative log-likelihood of masked code stacks:

$$\mathcal{L} = \mathbb{E}_{\mathbf{m} \sim q(\mathbf{m})} \left[\mathbb{E}_{\mathbf{S}} \left[\sum_{n : \mathbf{m}_n = 1} \sum_{d=1}^{D} - \log p_\theta(\mathbf{S}_{nd} | \mathbf{S}_{n,<d}, \mathbf{S}_{\setminus \mathbf{m}}) \right] \right]. \tag{11}$$

3.3 Draft-and-Revise: Two-Phase Decoding with Global Contexts of Generated Images

We propose a novel decoding algorithm, *Draft-and-Revise*, for Contextual RQ-Transformer to effectively generate high-quality images with diverse visual contents. We remark that our effective decoding method is required for bidirectional transformers to fully exploit the global contexts of images and achieve high performance of image generation. Here, we introduce the details of our Draft-and-Revise decoding and then explain how the two-phase decoding of Draft-and-Revise can effectively control the quality-diversity trade-off of generated images.

We define a partition $\mathbf{\Pi} = (\mathbf{m}^1, \cdots, \mathbf{m}^T)$ as a collection of pairwise disjoint T mask vectors to cover all spatial positions, where $\sum_{t=1}^{T} \mathbf{m}^t_n = 1$ for all $n \in [N]$. A partition $\mathbf{\Pi}$ is sampled from the distribution $p(\mathbf{\Pi}; T)$, which is the uniform distribution over all balanced partitions with size T:

$$p(\mathbf{\Pi}; T) = \text{Unif}\left(\{\mathbf{\Pi} = (\mathbf{m}^1, \cdots, \mathbf{m}^T) : |\mathbf{m}^t| = \tfrac{N}{T} \ \forall t \in [T]\}\right). \tag{12}$$

Table 1: FIDs, ISs, Precisions, and Recalls for class-conditional generation on ImageNet [7]. † denotes the use of pretrained classifier for rejection sampling, gradient guidance, or training.

	Params	$H \times W \times D$	FID↓	IS↑	Precision↑	Recall↑
BigGAN-deep [5]	112M	-	6.95	202.6	0.87	0.23
StyleGAN-XL† [32]	166M	-	2.3	262.1	0.78	0.53
ADM [9]	554M	-	10.94	101.0	0.69	0.63
ADM-G† [9]	608M	-	4.59	186.7	0.82	0.52
ImageBART [12]	3.5B	16×16×1	21.19	61.6	-	-
VQ-Diffusion [15]	518M	16×16×1	11.89	-	-	-
LDM-8 [30]	395M	32×32	15.51	79.03	0.65	0.63
LDM-8-G† [30]	506M	32×32	7.76	209.52	0.84	0.35
MaskGIT [6]	227M	16×16×1	6.18	182.1	0.80	0.51
VQ-GAN [13]	1.4B	16×16×1	15.78	74.3	-	-
RQ-Transformer [26]	1.4B	8×8×4	8.71	119.0	0.71	0.58
RQ-Transformer [26]	3.8B	8×8×4	7.55	134.0	0.73	0.58
RQ-Transformer† [26]	3.8B	8×8×4	3.80	323.7	0.82	0.50
Contextual RQ-Transformer	333M	8×8×4	5.45	172.6	0.81	0.49
Contextual RQ-Transformer	821M	8×8×4	3.45	221.9	0.82	0.52
Contextual RQ-Transformer	1.4B	8×8×4	3.41	224.6	0.79	0.54
Validation Data	-	-	1.62	234.0	0.75	0.67

We first define a procedure $\mathrm{UPDATE}(\mathbf{S}, \mathbf{\Pi})$ to update the sequence $\mathbf{S}$ as described in Algorithm 1, which updates $\mathbf{S}_n$ with $\mathbf{m}_n^t = 1$ for $t \in [T]$. Then, *Draft-and-Revise* decoding in Algorithm 2 generates a draft from the empty sequence of code stacks and improves the quality of the draft.

Draft phase In the draft phase, our model gradually infills the empty sequence of code stacks to generate a draft image, considering the global contexts of infilled code stacks. Let $\mathbf{S}^{\text{empty}}$ be an empty sequence of code stacks with $\mathbf{S}_n^{\text{empty}} = [\text{MASK}]^D$ for all n. Given a partition size T_{draft}, our model generates a draft image as

$$\mathbf{S}^{\text{draft}} = \mathrm{UPDATE}(\mathbf{S}^{\text{empty}}, \mathbf{\Pi}; \theta) \quad \text{where} \quad \mathbf{\Pi} \sim p(\mathbf{\Pi}; T_{\text{draft}}). \tag{13}$$

Revise phase The generated draft $\mathbf{S}^{\text{draft}}$ is repeatedly revised to improve the visual quality of the image, while preserving the overall structure of the draft. Given a partition size T_{revise} and the number of updates M, the draft $\mathbf{S}^0 = \mathbf{S}^{\text{draft}}$ is repeatedly updated M times as

$$\mathbf{S}^m = \mathrm{UPDATE}(\mathbf{S}^{m-1}, \mathbf{\Pi}; \theta) \quad \text{where} \quad \mathbf{\Pi} \sim p(\mathbf{\Pi}; T_{\text{revise}}) \quad \text{for } m = 1, \cdots, M. \tag{14}$$

Note that Draft-and-Revise is not a tailored method, since we can adopt any mask-infilling-based generation method [4, 6] for UPDATE in Algorithm 1. For example, confidence-based decoding [6, 16], which iteratively updates $\mathbf{S}$ from high-confidence to low-confidence predictions, can be used for UPDATE. However, we find that confidence-based decoding generates low-diversity images with oversimplified contents, since a model tends to predict simple visual patterns with high confidence. In addition, confidence-based decoding often leads to biased unmasking patterns, which are not used in training. We attach the detailed discussion about confidence-based decoding in Appendix C. Thus, we use a uniformly random partition $\mathbf{\Pi}$ in UPDATE as the most simplified rule, leaving investigations on sophisticated update methods as future work.

We postulate that our Draft-and-Revise can generate high-quality images with diverse contents by explicitly dividing two phases. Specifically, a model first generates draft images with diverse visual contents despite the rather low quality of drafts. After semantically diverse images are generated as drafts, we use sampling strategies such as temperature scaling [19] and classifier-free guidance [20] in the revise phase to improve the visual quality of the drafts, while preserving the major semantic contents in drafts. Thus, our method can improve the performance of image generation by effectively controlling the quality-diversity trade-off. In addition, we emphasize that the two-phased decoding is intuitive and resembles the image generation process of human experts, who repeatedly refine their works to improve the quality after determining the overall contents first.

Figure 3: The examples of generated 256×256 images of our model trained on (Top) ImageNet and (Bottom) CC-3M. The used text conditions are "Sunset over the skyline of a {beach, city}.", "an avocado {in the desert, on the seashore}.", and "a painting of a {dog, cat} with sunglasses.".

4 Experiments

In this section, we show that our Draft-and-Revise with Contextual RQ-Transformer can outperform previous approaches for class- and text-conditional image generation. In addition, we conduct an extensive ablation study to understand the effects of Draft-and-Revise decoding on the quality and diversity of generated images, and the sampling speed. We use the publicly released RQ-VAE [26] to represent a 256×256 resolution of images as 8×8×4 codes. For a fair comparison, we make Contextual RQ-Transformer have the same model size as the previous RQ-Transformer [26]. For training, the quarter-period of cosine is used as the mask scheduling function γ in Eq. 10 following the previous studies [6, 27]. We include the implementation details in Appendix.

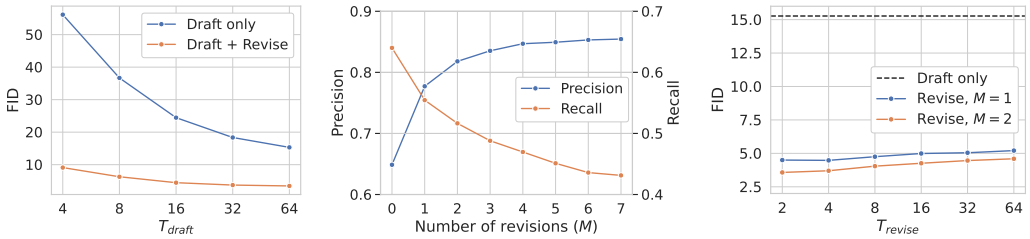
4.1 Class-conditional Image Generation

We train Contextual RQ-Transformer with 333M, 821M, and 1.4B parameters on ImageNet [7] for class-conditional image generation. For Draft-and-Revise decoding, we use $T_{\text{draft}} = 64$, $T_{\text{revise}} = 2$, and $M = 2$. We use temperature scaling [19] and classifier-free guidance [20] only in the revise phase, while none of the strategies are applied in the draft phase. Fréchet Inception Distance (FID) [18], Inception Score (IS) [31], and Precision and Recall [25] are used for evaluation measures.

Table 1 shows that Contextual RQ-Transformer significantly outperforms the previous approaches. Notably, Contextual RQ-Transformer with 333M parameters outperforms RQ-Transformers with 1.4B and 3.8B parameters on all evaluation measures, despite having only about 4.2× and 11.4× fewer parameters. In addition, the performance is improved as the number of parameters increases to 821M and 1.4B. Our model with 333M parameters is competitive with MaskGIT [6], and hence we conduct additional analysis on the effect of confidence-based decoding in Appendix C. Contextual RQ-Transformer can achieve the lower FID score without a pretrained classifier than ADM-G and 3.8B parameters of RQ-Transformer with the use of pretrained classifier. StyleGAN-XL also uses a pretrained classifier during both training and image generation and achieves the lowest FID in Table 1. However, our model with 1.4B parameters has higher precision and recall than StyleGAN-XL, implying that our model generates images of better fidelity and diversity without a pretrained classifier. Our high performance without a classifier is remarkable, since the gradient guidance and rejection sampling are the tailored techniques to the model-based evaluation metrics in Table 1. Considering that the performance is marginally improved as the number of parameters increases from 821M to 1.4B, an improved RQ-VAE can boost the performance of Contextual RQ-Transformer, since the reconstruction quality determines the best results of generated images.

4.2 Text-conditional Image Generation

We train our model with 333M and 654M parameters on CC-3M [35] for text-to-image (T2I) generation as described in Appendix A. A text condition is encoded into 32 tokens using Byte Pair Encoding [34, 37]. We report CLIP-score [28] with ViT-B/32 [11] as the correspondence between texts and images.



(a) (b) (c)

Figure 4: Ablation study of Draft-and-Revise decoding. (a) FIDs with the partition sizes of the draft phase T_{draft}. (b) Precision and recall with the revision iterations M. (c) FID with T_{revise}.

Figure 5: Examples of generated images in the draft phase (left) and revise phases at each iteration $M = 1, 2, 3, 4, 5$. The draft images are generated with $T_{\text{draft}} = 8$ (top) and $T_{\text{draft}} = 64$ (bottom).

Contextual RQ-Transformer in Table 2 outperforms the previous T2I generation models. Contextual RQ-Transformer with 333M parameters achieves better FID than RQ-Transformer with 654M parameters, and outperforms ImageBART and LDM-4, although our model has 12× fewer parameters than ImageBART. When we increase the number of parameters to 654M, our model achieves state-of-the-art FID on CC-3M. In Figure 3, our model generates images with unseen texts in CC-3M.

Table 2: FIDs and CLIP scores [28] on the validation dataset of CC-3M [35] for T2I generation.

	Params	FID↓	CLIP-s↑
VQ-GAN [13]	600M	28.86	0.20
ImageBART [12]	2.8B	22.61	0.23
LDM-4 [30]	645M	17.01	0.24
RQ-Transformer [26]	654M	12.33	0.26
Ours	333M	10.44	0.26
Ours	654M	9.80	0.26

4.3 Conditional Image Inpainting

We conduct conditional image inpainting where a model infills a masked area according to the given condition and contexts. Figure 1 shows the example of image inpainting by RQ-Transformer (middle) and Contextual RQ-Transformer (right), when the class-condition is *school bus*. RQ-Transformer cannot attend to the right and bottom sides of the masked area and fails to generate a coherent image with given contexts. However, our model can complete the image to be coherent with given contexts by exploiting global contexts. We attach more examples of image inpainting in Appendix.

4.4 Ablation Study on Draft-and-Revise

We conduct an extensive ablation study to demonstrate the effectiveness of Draft-and-Revise decoding of our framework. We use Contextual RQ-Transformer with 821M parameters trained on ImageNet.

Quality improvement of draft images in the revise phase Figure 4(a) shows the effects of T_{draft} on draft images and their quality improvement in the revised phase with $T_{\text{revise}} = 2$ and $M = 2$.

In the draft phase, FID is improved as T_{draft} increases from 4 to 64. At each inference, Contextual RQ-Transformer generates N/T_{draft} code stacks in parallel, starting with the empty sequence. Thus, the model with a large T_{draft} generates a small number of code stacks at each inference and can avoid generating incoherent code stacks in the early stage of the draft phase. Although FIDs in the draft phase are worse due to low precision, we intend to increase the diversity of generated images, since the precision is significantly improved in the revise phase as shown in Figure 4(b).

Effect of revision iteration M and partition size T_{revise} in the revise phase Figure 4(b) shows the effects of the number of updates M in the revise phase on the quality and diversity of generated images. Since the quality-diversity trade-off exists as the updates are repeated, we select $M = 2$ as the default hyperparameter to balance the precision and recall, considering that the increase of precision starts to slow down. Interestingly, Figure 5 shows that the overall contents remain unchanged even after $M > 2$. Thus, we claim that Draft-and-Revise decoding does not harm the perceptual diversity of generated images throughout the revise phase despite the consistent deterioration of recall.

Figure 4(c) shows the effects of T_{revise} on the quality of generated images. The FIDs are significantly improved in the revise phase regardless of the choice of T_{revise}, but increasing T_{revise} slightly deteriorates FIDs. We remark that some code stacks of a draft can be erroneous due to its low quality, and a model with large T_{revise} slowly updates a small number of code stacks at once in the revise phase. Therefore, the updates with large T_{revise} can be more influenced by the erroneous code stacks. Although $T_{\text{revise}} = 2$ updates half of an image at once, our draft-and-revise decoding successfully improves the quality of generated images, while preserving the global contexts of drafts, as shown in Figure 5. The study on self-supervised learning [17] also reports similar results, where a masked auto-encoder reconstructs the global contexts of an image after masking half of the image.

Quality-diversity control of Draft-and-Revise Our *Draft-and-Revise* decoding can effectively control the quality-diversity trade-off in generated images. Table 3 shows FID, precision (P), and recall (R) according to the use of classifier-free guidance [20] with a scale of 1.8, while applying temperature scaling with 0.8 only to the revise phase. Contextual RQ-Transformer without the guidance already outperforms RQ-Transformer with 3.8B parameters and demonstrates the effectiveness of our framework. When the guidance is used for both draft and revise phases, the precision dramatically increases but the recall decreases to 0.33. Consequently, FID becomes worse due to the lack of diversity in generated images. However, when the guidance is applied only to the revise phase, our model achieves the lowest FID, as the quality and diversity are well-balanced. Thus, the explicitly separated two phases of Draft-and-Revise can effectively control the issue of quality-diversity trade-off by generating diverse drafts and then improving their quality.

Table 3: The effects of classifier-free guidance on the image generation.

Draft	Revise	FID	P	R
		5.78	0.72	0.58
	✓	3.45	0.82	0.52
✓	✓	8.90	0.92	0.33

Trade-off between quality and sampling speed Table 4 shows that our framework can control the trade-off between the quality and sampling speed according to T_{draft} after we fix $T_{\text{revise}} = 2$ and $M = 2$. We generate 5,000 samples with batch size of 100 as the previous study [26]. Contextual RQ-Transformer with $T_{\text{draft}} = 8$ outperforms both FID and sampling speed of VQGAN and RQ-Transformer with 1.4B parameters. Although the sampling speed becomes slow with increased T_{draft}, the FID scores are consistently improved. We remark that the sampling speed with $T_{\text{draft}} = 64$ is about 3× slower than RQ-Transformer, but our model outperforms 3.8B parameters of RQ-Transformer with rejection sampling in Table 1. The results show that our framework has inexpensive costs to generate high-quality images, considering that rejection sampling requires generating up to 20× more samples.

Table 4: Comparison of FID and the sampling speed of image generation.

	FID	s/sample
VQGAN	15.78	0.16
RQ-Transformer	8.71	0.04
Contextual RQ-Transformer		
$T_{\text{draft}} = 8$	5.41	0.03
$T_{\text{draft}} = 32$	3.73	0.06
$T_{\text{draft}} = 64$	3.45	0.10

5 Conclusion

In this study, we have proposed *Draft-and-Revise* for an effective image generation framework with Contextual RQ-Transformer. After an image is represented as a sequence of code stacks, Contextual RQ-Transformer is trained to infill a randomly masked sequence. Then, Draft-and-Revise decoding is used to generate high-quality images by first generating a draft image with diverse contents and then improving its visual quality based on the global contexts of the draft. Consequently, we can achieve state-of-the-art results on ImageNet and CC-3M, demonstrating the effectiveness of our framework.

Our study has two main limitations to be further explored. Firstly, Draft-and-Revise decoding always updates all code stacks in the revise phase, although some code stacks might not need an update. In future work, a selective method can be developed to improve the efficiency of the revise phase by a sophisticated approach. Secondly, our generative model is not validated on various downstream tasks. Since masked token modeling is successful self-supervised learning for texts [8] and images [3, 17], a unified model for both generative and discriminative tasks [24] is worth exploration for future work.

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, 35%; No.2021-0-00537: Visual Common Sense, 30%) and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2021R1A2B5B03001551, 35%)

References

[1] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEit: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022.

[4] Sam Bond-Taylor, Peter Hessey, Hiroshi Sasaki, Toby P Breckon, and Chris G Willcocks. Unleashing transformers: Parallel token prediction with discrete absorbing diffusion for fast high-resolution image generation from vector-quantized codes. *arXiv preprint arXiv:2111.12701*, 2021.

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

[6] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[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.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[9] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34, 2021.

[10] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34, 2021.

[11] 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.

[12] Patrick Esser, Robin Rombach, Andreas Blattmann, and Bjorn Ommer. Imagebart: Bidirectional context with multinomial diffusion for autoregressive image synthesis. *Advances in Neural Information Processing Systems*, 34, 2021.

[13] 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.

[14] Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6112–6121, 2019.

[15] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[16] Ligong Han, Jian Ren, Hsin-Ying Lee, Francesco Barbieri, Kyle Olszewski, Shervin Minaee, Dimitris Metaxas, and Sergey Tulyakov. Show me what and tell me how: Video synthesis via multimodal conditioning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[17] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[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.

[19] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

[20] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[21] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2019.

[22] 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.

[23] 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.

[24] Saehoon Kim, Sungwoong Kim, and Juho Lee. Hybrid generative-contrastive representation learning. *arXiv preprint arXiv:2106.06162*, 2021.

[25] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[26] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[27] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[28] 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.

[29] 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.

[30] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[31] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[32] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. *arXiv preprint arXiv:2202.00273*, 2022.

[33] Nikolay Savinov, Junyoung Chung, Mikolaj Binkowski, Erich Elsen, and Aaron van den Oord. Step-unrolled denoising autoencoders for text generation. In *International Conference on Learning Representations*, 2022.

[34] 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.

[35] 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.

[36] 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.

[37] 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, October 2020. Association for Computational Linguistics.

[38] Zhu Zhang, Jianxin Ma, Chang Zhou, Rui Men, Zhikang Li, Ming Ding, Jie Tang, Jingren Zhou, and Hongxia Yang. UFC-BERT: Unifying multi-modal controls for conditional image synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Our code is included in the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Sec. 4 and Appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We used publicly available datasets and cited them in Sec. 4.
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Our code is included in the supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Implementation Details

A.1 Details of RQ-VAE

RQ-VAE [26] is a generalized version of VQ-VAE [36] and VQGAN [13], since RQ-VAE with $D = 1$ is equivalent to VQ-VAE or VQGAN. When $D > 1$, RQ-VAE recursively discretizes the feature map of an image for a precise approximation of the feature map using the codebook. When the codebook size of RQ is K, RQ with depth D is as capable as VQ with K^D size of a codebook, since RQ can represent at most K^D clusters in a vector space. That is, if the codebook sizes are the same, RQ with $D > 1$ can approximate a feature vector more accurately than VQ. Thus, RQ-VAE can further reduce the spatial resolution of code map than VQ-VAE and VQ-GAN, and therefore outperforms previous autoregressive models with discrete representations.

Following the previous studies [13, 26, 36], the training of RQ-VAE uses the reconstruction loss, the commitment loss, the adversarial training [22], and the LPIPS perceptual loss [23]. The codebook $\mathcal{C}$ of RQ-VAE is updated using the exponential moving average during training [26, 36].

In experiments, we use the pretrained RQ-VAE, which is publicly available[3]. The RQ-VAE uses the codebook size of 16,384 to represent a 256×256 resolution of an image as 8×8×4 shape of a code map. The architecture of RQ-VAE is the same as VQGAN [13] except for adding residual blocks in the encoder and the decoder to reduce the spatial resolution of the code map more than VQGAN.

A.2 Architecture of Contextual RQ-Transformer

Table 5 summarizes the architecture details of Contextual RQ-Transformers to be trained on ImageNet and CC-3M. Contextual RQ-Transformer consists of two compartments: *bidirectional spatial transformer* with N_{spatial} self-attention blocks and *depth transformer* with N_{depth} causal self-attention blocks. The dimensionality of embeddings in multi-headed self-attentions is denoted d_{model}, while the dimensionality for each attention head is 64.

Table 5: Architecture details of Contextual RQ-Transformer for ImageNet and CC-3M.

Dataset	# params.	N_{spatial}	N_{depth}	d_{model}
ImageNet	371M	24	4	1024
	821M	24	4	1536
	1.4B	42	6	1536
CC-3M	366M	21	4	1024
	654M	26	4	1280

A.3 Training details

All Contextual RQ-Transformers are trained with AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and weight decay 0.0001. Each model is trained for 300 epochs with the cosine learning rate schedule with the initial value of 0.0001 and the final value of 0, for both ImageNet and CC-3M. We use eight NVIDIA A100 GPUs to train the models with 821M and 1.4B parameters on ImageNet and the model with 650M parameters on CC-3M, while four GPUs are used for the models with 366M parameters. For our model with 821M and 1.4B parameters on ImageNet, the training takes at most 10 days.

A.4 Draft-and-Revise decoding details

We use temperature scaling with the 0.8 scale in the revise phase and do not apply the temperature scaling in the draft phase. The classifier-free guidance is also applied only to the sampling in the revise phase. We use 1.4, 1.8, and 2.0 scales of guidance for 371M, 821M, and 1.4B parameters of Contextual RQ-Transformer on ImageNet, respectively. In addition, (M, T_{revise}) is (3,2), (2,2), and (2,2), respectively. In the revise phase for Contextual RQ-Transformer on CC-3M, we use the 1.1 scale of classifier-free guidance with $(M, T_{\text{revise}}) = (2, 4)$.

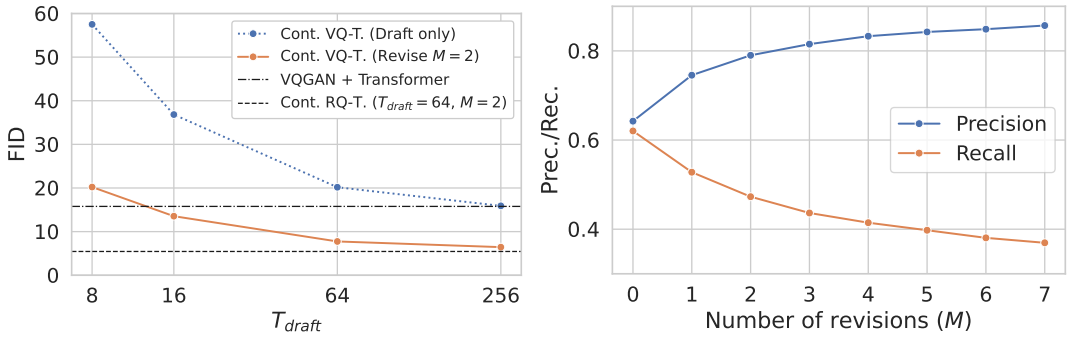
B The Compatibility of Draft-and-Revise with 16×16 VQGAN

Since RQ is a generalized VQ, our framework of Draft-and-Revise with Contextual RQ-Transformer is also applicable to VQ-VAE [36] and VQGAN [13]. Note that RQ-VAE with $D = 1$ is equivalent to VQGAN, where their spatial resolutions of code maps are the same. To validate that our framework

[3] `https://github.com/kakaobrain/rq-vae-transformer`

Table 6: Performance of Contextual RQ-Transformers on 16×16×1 RQ-VAE.

	# params	$H \times W \times D$	FID	P	R	s/sample
VQ-GAN [13]	1.4B	16×16×1	15.78	-	-	-
RQ-Transformer [26]	1.4B	8×8×4	8.71	0.71	0.58	0.04
Contextual VQ-Transformer	350M	16×16×1	6.44	0.79	0.47	0.83
Contextual RQ-Transformer	371M	8×8×4	5.45	0.81	0.49	0.08



(a) (b)

Figure 6: Effect of Draft-and-Revise decoding on Contextual VQ-Transformer. (a) FID subject to T_{draft}. (b) Precision and recall subject to M.

is also effective on the 16×16 shape of code map by VQGAN, we first train an RQ-VAE, which represents an image as 16×16×1 shape of code maps and has the identical architecture to VQGAN with 16×16 shape of code map. Then, we train a Contextual RQ-Transformer with 350M parameters on 16×16×1 codes. We notate the Contextual RQ-Transformer with $D = 1$ as *Contextual VQ-Transformer* throughout this section.

Figure 6 shows that the Draft-and-Revise decoding also generalizes to Contextual VQ-Transformer and can control the quality-diversity trade-off in the same manner as with the Contextual RQ-Transformer. We fix $T_{\text{revise}} = 2$ and $M = 2$, and use temperature scaling of 0.8 and classifier-free guidance of 2.4 only in the revise phase. As shown in Figure 6(a), the best performance is achieved when $T_{\text{draft}} = 256$ and the corresponding FID, precision (P), and recall (R) are 6.44, 0.79, and 0.47, respectively, as reported in Table 6. Note that Contextual VQ-Transformer outperforms the AR model such as VQGAN, although our model has $4\times$ fewer parameters than the AR model. In addition, our draft-and-revise decoding with Contextual VQ-Transformer also works as well as Contextual RQ-Transformer, showing that the iterative updates in the revise phase consistently increase precisions and decreases recalls in Figure 6(b). Consequently, the results validate that our framework is more effective for high-quality image generation than AR modeling.

Although our framework is compatible with VQGAN, we emphasize that Contextual RQ-Transformer is more effective than Contextual VQ-Transformer. Contextual RQ-Transformer with a similar number of parameters outperforms the Contextual VQ-Transformer in terms of FID, precision, and recall in Table 6. In addition, Contextual RQ-Transformer has about $10\times$ faster speed for image generation than Contextual VQ-Transformer, since the computational complexity of self-attention is mainly determined by the sequence length. Although the comparison of FID, precision, and recall is not entirely fair due to the different performance between RQ-VAE and VQGAN, Contextual RQ-Transformer can achieve state-of-the-art performance with lower computational costs than Contextual VQ-Transformer. Thus, masked modeling in RQ representations is more effective and efficient than in VQ representations, if equipped with our Contextual RQ-Transformer.

C Comparison with Confidence-based Mask-infilling Strategies

In this section, we examine how the selection of UPDATE in Algorithm 1 affects the performance of image generation. First, instead of using our random updates of spatial positions, we consider

Table 7: Comparison of confidence-based mask-infilling strategies and our random partitioning strategy in the draft phase.

	Draft Images			Revised Images		
	FID	P	R	FID	P	R
$T_{\text{draft}} = 8$						
Top-C	18.28	0.67	0.43	11.92	0.79	0.35
Top-C-50%	13.26	0.71	0.53	6.13	0.86	0.41
Random	36.68	0.58	0.60	6.28	0.80	0.49
$T_{\text{draft}} = 64$						
Top-C	34.07	0.54	0.42	27.88	0.62	0.37
Top-C-50%	7.22	0.75	0.53	6.84	0.84	0.42
Random	15.32	0.65	0.64	3.45	0.82	0.52

a confidence-based mask-infilling strategy of MaskGIT [6] and denote the sampling strategy of MaskGIT as *Top-C*. At each inference step, *Top-C* considers the predicted confidences and determines the unmasked positions to have highly confident predictions. Then, we also consider a mixed strategy, *Top-C-50%*, which first filters out the bottom 50% confident positions, and then randomly selects the unmasked positions among the positions with top 50% high confidence. *Top-C-50%* is similar to the combination of random sampling after top-k or top-p [21] filtering. We denote our mask-infilling strategy as *Random*, which randomly determines the unmasked regions at each inference in the draft phase. We use Contextual RQ-Transformer with 821M parameters and fix the parameters of Draft-and-Revise decoding to $T_{\text{draft}} = 64$, $T_{\text{revise}} = 2$, and $M = 2$ and apply temperature scaling of 0.8 and classifier-free guidance of 1.8 in the revise phase. We report FID, precision (P), and recall (R) of the generated images in the draft and revise phases.

Table 7 shows the effect of Top-C, Top-C-50%, and Random in Draft-and-Revise decoding. Regardless of the selection of mask-infilling strategies in the draft phase, our draft-and-revise decoding consistently improves the performance of image generation after the revise phase. The results imply that our framework of Draft-and-Revise can be effectively generalized to various approaches of mask-infilling-based image generation.

When $T_{\text{draft}} = 8$, the draft images of Top-C have better FID but worse recall than the draft images of Random. Nonetheless, the quality of the draft images is significantly improved with the revise phase and subsequently Random outperforms Top-C in all three metrics. Top-C-50% achieves the best FID in both the draft and revised images, but the recall of Top-C-50% is significantly lower than the recall of Random. When $T_{\text{draft}} = 64$, the draft images of Top-C exhibit worse metrics than the draft images of Random. Although Top-C-50% achieves lower FID than Top-C and Random in terms of draft images, Random outperforms Top-C and Top-C-50% after the revise phase. This supports our claim that Draft-and-Revise decoding better controls quality-diversity trade-off when drafts are generated with maximal diversity.

By the visual analysis in Figure 7, we find that the limited performance of confidence-based mask-infilling strategies results from the bias of a model on high-confident predictions. That is, Top-C tends to predict only simple patterns in the early phase of mask-infilling, since the simple visual patterns are prone to have high-confident predictions. This effect becomes more apparent when $T_{\text{draft}} = 64$ as only the code stack with the highest confidence is included in the sample at each inference, thereby the diversity of the generated images becomes severely limited. Indeed, it is shown in Figure 7 that the confidence-based methods first infill the backgrounds with simple visual patterns, and the resulting samples exhibit low diversity of visual contents. Due to the limited diversity, the confidence-based mask-infilling strategies limit the performance of FID, although our draft-and-revise decoding further improves the visual quality of generated images after the revise phase. Thus, we conclude that our sampling strategy, which first generates diverse visual contents and then improves their quality, is effective for draft-and-revise decoding to achieve high performance of image generation.

D Additional Generation Examples

In this section, we show additional examples of generated images by our Contextual RQ-Transformer. We use 1.4B parameters of Contextual RQ-Transformer trained on ImageNet for class-conditional

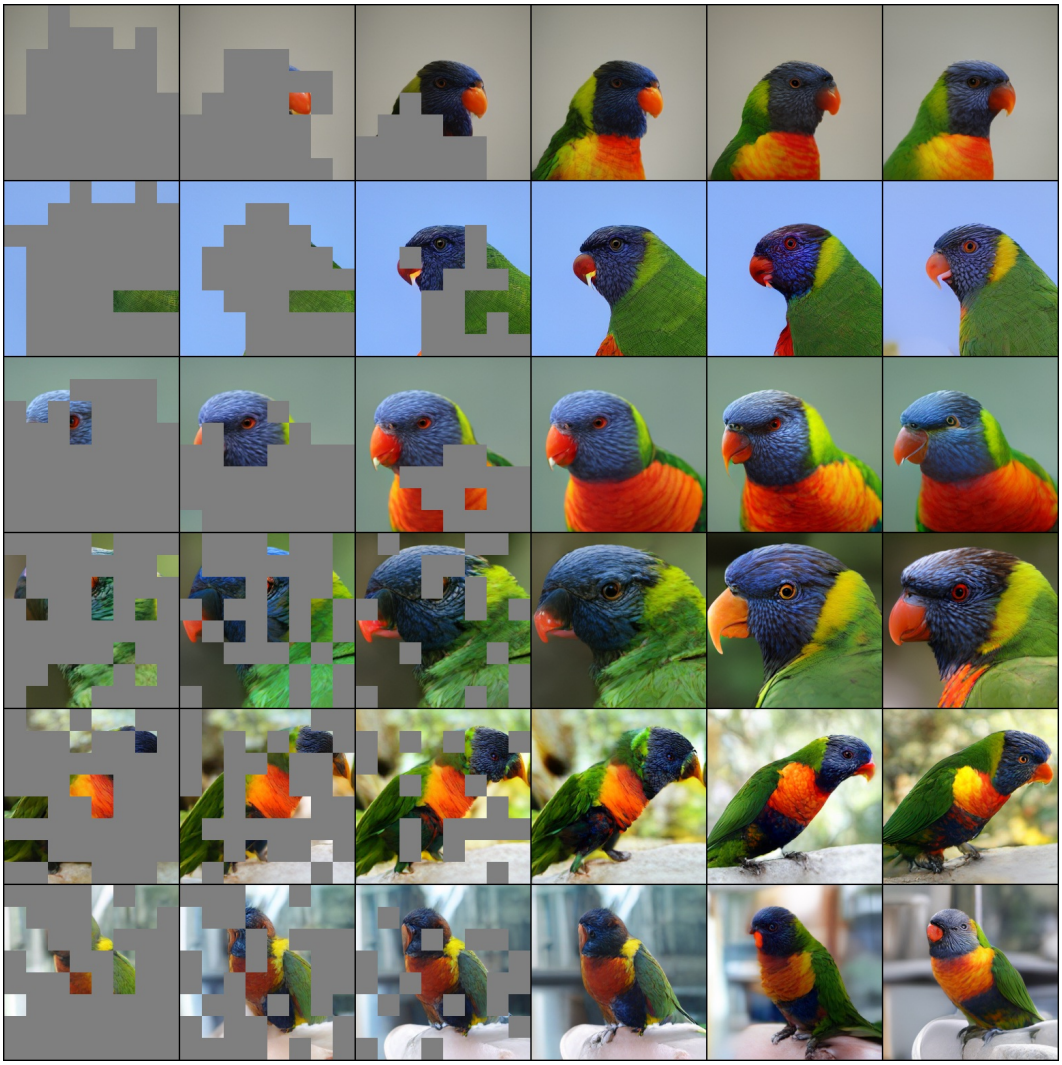


Figure 7: Intermediary samples of mask-infilling strategies with $T_{\text{draft}} = 64$ and subsequently revised samples. From left to right, the first four images are the samples with the corresponding masking pattern of every 16th mask-infilling step, and the last two images are the revised samples for $M = 1, 2$. (Top 3 rows) Mask-infilled with a confidence-based strategy, *Top-C*. (Bottom 3 rows) Mask-infilled with our strategy, *Random*.

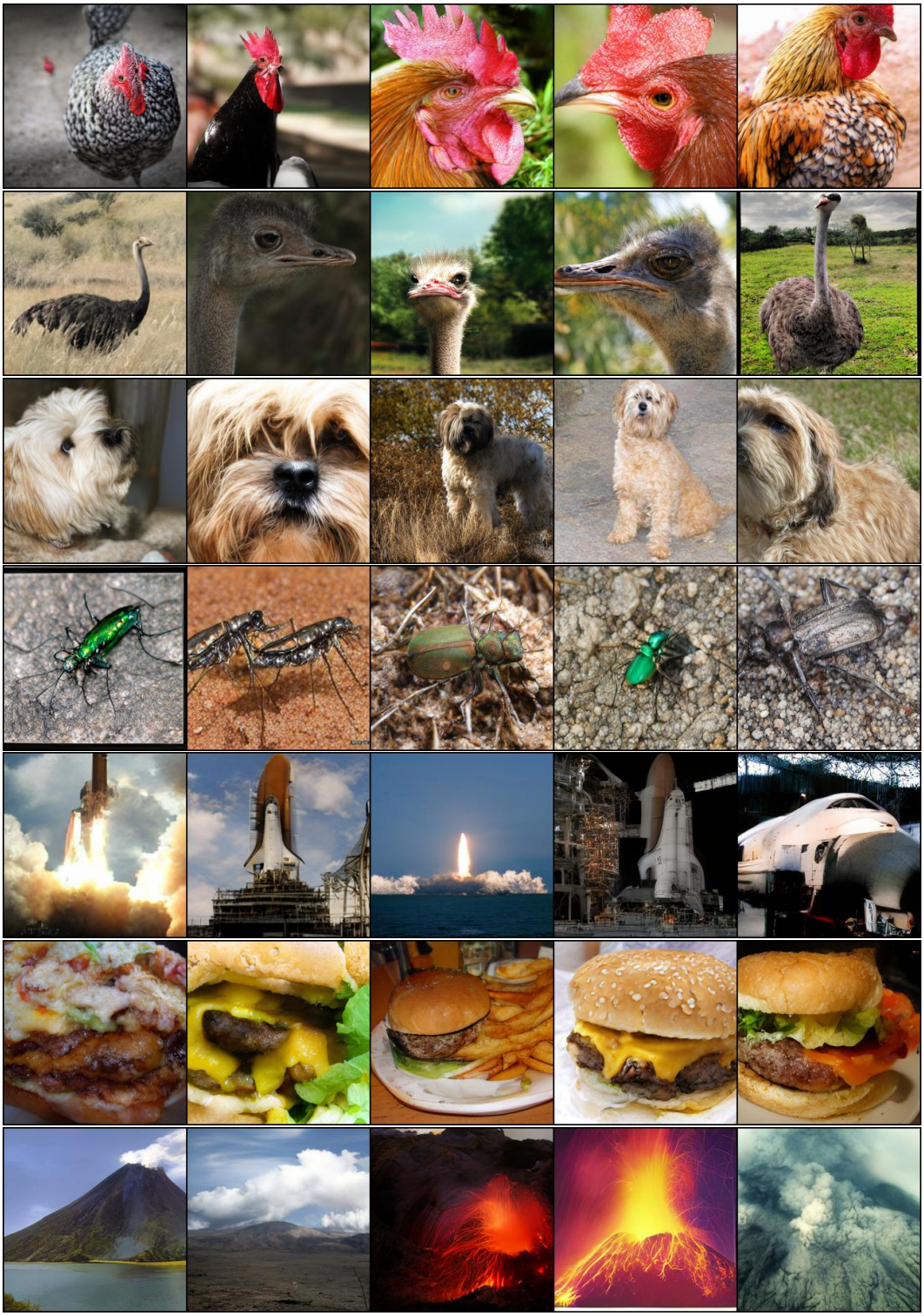


Figure 8: Additional examples of class-conditional image generation by our model with 1.4B parameters trained on ImageNet. The class conditions are Cock (7), Ostrich (9), Tibetan terrier (200), Space shuttle (812), Cheeseburger (933), and Volcano (980), respectively.

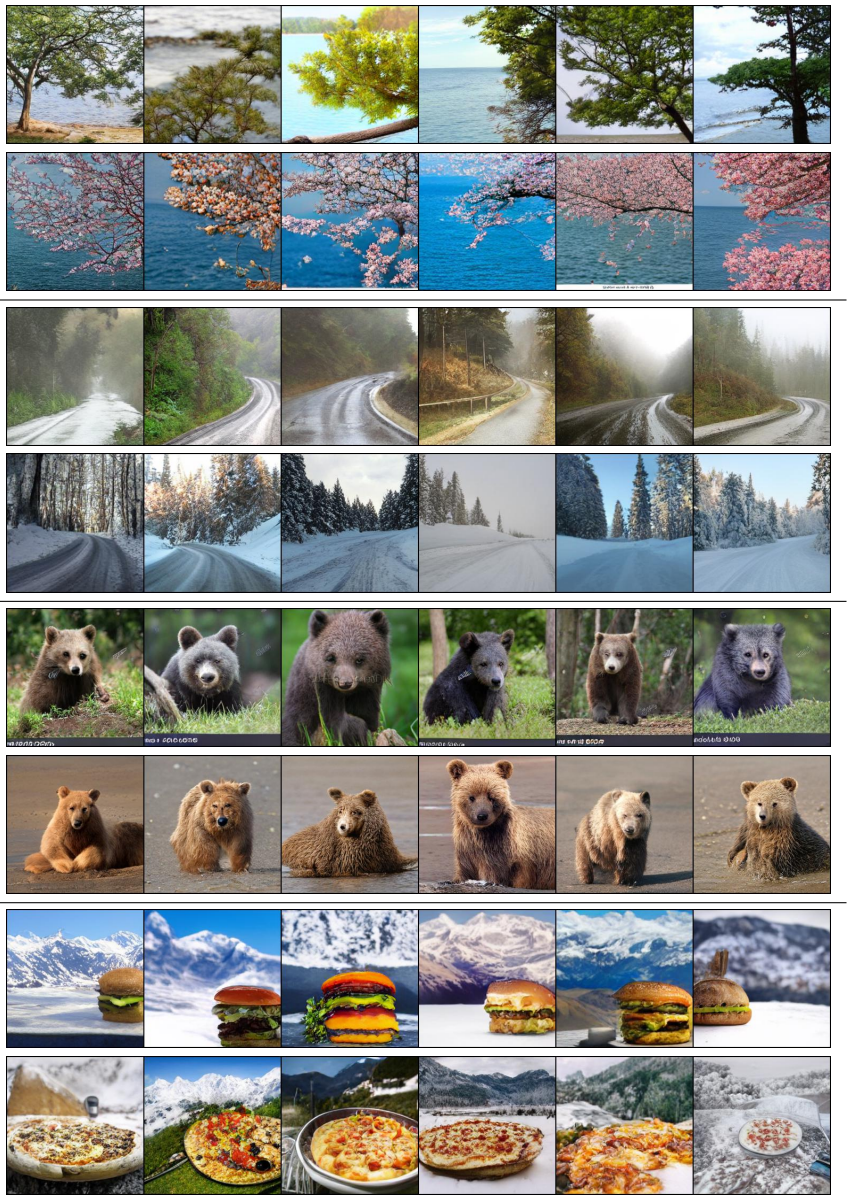


Figure 9: Additional examples of text-conditional image generation by our model with 654M parameters trained on CC-3M. † denotes that the caption exists in the validation dataset of CC-3M.

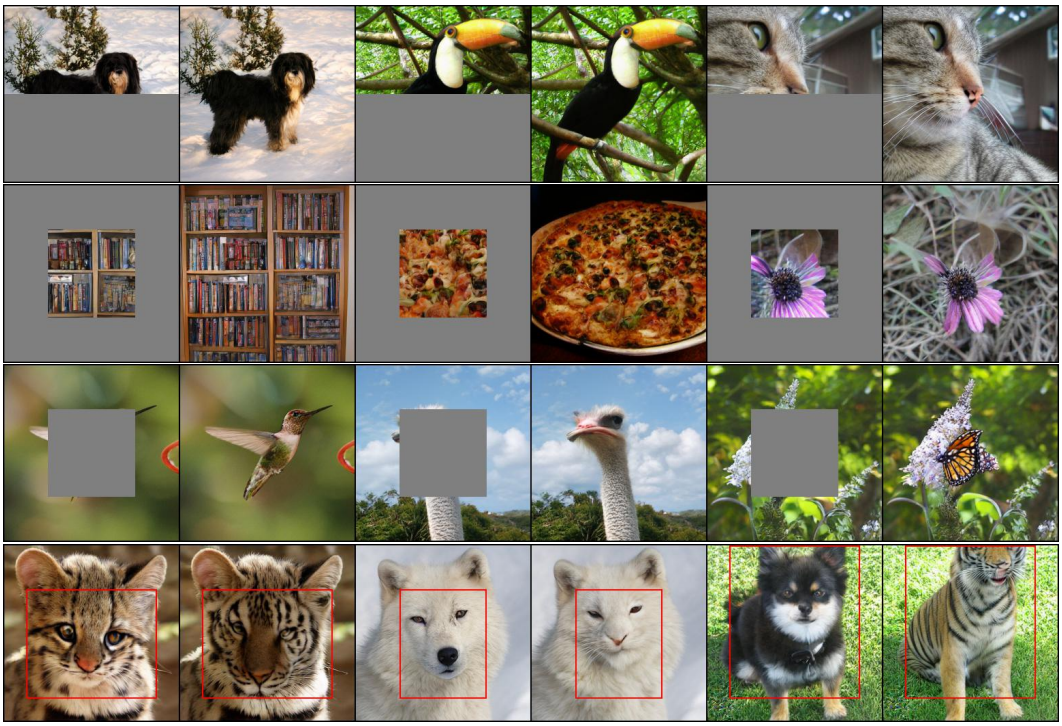


Figure 10: Additional examples of image inpainting by our model with 1.4B parameters trained on ImageNet. All masked images are taken from the validation set of ImageNet. (Top 3 rows) Inpainted images when conditioned on the class of the original image. (Bottom row) Images where the region inside the red box is inpainted with the class condition Tiger (292).

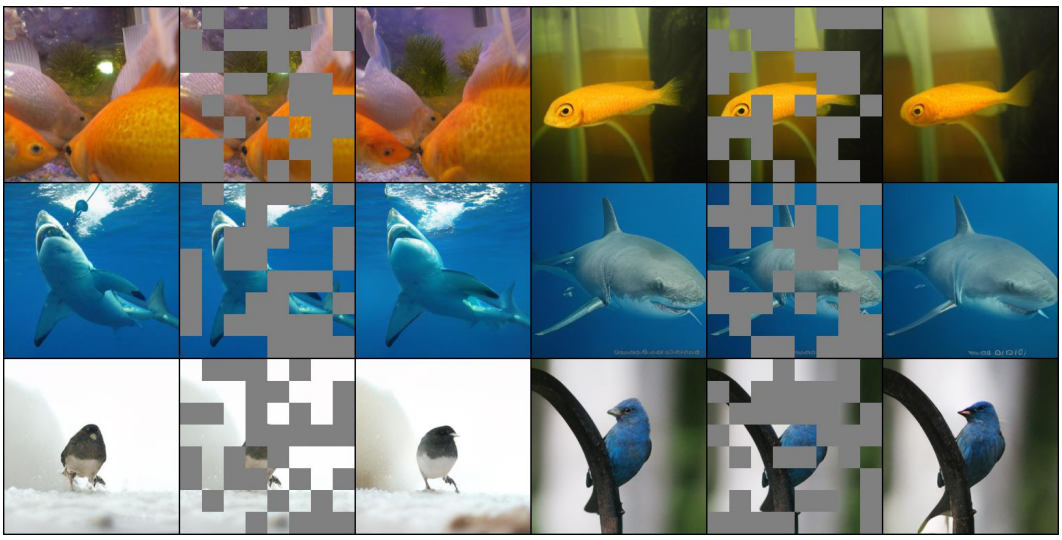


Figure 11: The triplets of the original image, randomly masked image, and mask-infilled image by our model. Although half of an image is randomly masked, Contextual RQ-Transformer can infill the masks and remains the global contexts of the original image.

image generation and 654M parameters of our model trained on CC-3M for text-conditional image generation.

D.1 Examples of Class-conditional Image Generation

Figure 8 shows the generated images on class conditions in ImageNet. Contextual RQ-Transformer generates diverse and high-quality images conditioned on given class conditions.

D.2 Examples of Text-conditioned Image Generation

Figure 8 shows the generated images by our model on various text conditions. We use the captions in the validation dataset[†] of CC-3M and change a keyword in the caption to validate the generalization power of our model on unseen compositions of texts. Figure 8 shows that our Contextual RQ-Transformer can generate high-quality images on diverse compositions of text conditions, even though the text condition is unseen during training.

D.3 Examples of Image In-painting

Figure 10 shows that Draft-and-Revise decoding can also be used to inpaint the prescribed region of a given image. For image inpainting, a random partition $\mathbf{\Pi}$ used in each application of UPDATE is set to be the partition of the masked region, instead of all spatial positions. The first three rows of Figure 10 show the image inpainting results, where the original images are taken from the validation set of ImageNet, either bottom, center, or outside is masked, and the class of the original image is given as a condition. On the other hand, the last row of Figure 10 shows that our Contextual RQ-Transformer is also capable of image editing via inpainting, by conditioning on a class-condition that is not the class of the original image.

D.4 Examples of Mask-Infilling of Half Masked Images

After we randomly mask the half of images in the ImageNet validation dataset, Contextual RQ-Transformer infills the masked regions in Figure 11. The results show that Contextual RQ-Transformer can infill the masked regions, while preserving the global contexts of original images. Note that the results are also aligned to the experimental results in previous approaches [17, 6]. That is, our draft-and-revise decoding can preserve the global contexts of the draft images in the revise phase, although we use a small value of T_{revise} such as 2 or 4. Note that the fine-grained details can be changed after infilling masks, since our method randomly samples the codes for unmasking based on the predicted softmax distribution over codes in Eq. 9.

D.5 Societal Impact

Since our framework can learn large-scale datasets for a real-world applications, the model has a potential of negative social impacts by generating socially biased or violent images according to the training data. Thus, exhaustive filtering of training data and evaluation after training have to be conducted for avoiding social problems.