# EFFICIENT GENERATIVE MODELING WITH RESIDUAL VECTOR QUANTIZATION-BASED TOKENS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

We explore the use of Residual Vector Quantization (RVQ) for high-fidelity generation in vector-quantized generative models. This quantization technique maintains higher data fidelity by employing more in-depth tokens. However, increasing the token number in generative models leads to slower inference speeds. To this end, we introduce ResGEN, an efficient RVQ-based discrete diffusion model that generates high-fidelity samples without compromising sampling speed. Our key idea is a direct prediction of vector embedding of collective tokens rather than individual ones. Moreover, we demonstrate that our proposed token masking and multi-token prediction method can be formulated within a principled probabilistic framework using a discrete diffusion process and variational inference. We validate the efficacy and generalizability of the proposed method on two challenging tasks across different modalities: conditional *image generation* on ImageNet 256×256 and zero-shot *text-to-speech synthesis*. Experimental results demonstrate that ResGEN outperforms autoregressive counterparts in both tasks, delivering superior performance without compromising sampling speed. Furthermore, as we scale the depth of RVQ, our generative models exhibit enhanced generation fidelity or faster sampling speeds compared to similarly sized baseline models. The project page can be found at `https://x8cg6mhs1qtf.github.io`.

## 1 INTRODUCTION

Recent advancements in deep generative models have shown significant success in high-quality, realistic data generation across multiple domains, including language modeling (Achiam et al., 2023; Touvron et al., 2023; Reid et al., 2024), image generation (Rombach et al., 2022; Saharia et al., 2022; Betker et al., 2023), and audio synthesis (Wang et al., 2023; Shen et al., 2023; Rubenstein et al., 2023). While these models have demonstrated remarkable success, particularly with the effective scaling with both data size and model size (Kaplan et al., 2020; Peebles & Xie, 2023), challenges remain when aiming for high-fidelity generation, especially in terms of balancing generation quality with computational efficiency. The demand for more detailed, high-resolution outputs such as images (Kang et al., 2023; He et al., 2023), videos (Bar-Tal et al., 2024) and audio (Evans et al., 2024; Copet et al., 2024), has led to the exploration of new approaches that can handle long input sequences and complex data structure effectively (Saharia et al., 2022; Ding et al., 2023).

One promising approach to address these challenges is Residual Vector Quantization (RVQ) (Chen et al., 2010), which improves data reconstruction quality without increasing sequence length. RVQ extends Vector Quantized Variational Autoencoders (VQ-VAEs) (Van Den Oord et al., 2017) by iteratively applying vector quantization to the residuals of previous quantizations (Lee et al., 2022; Zeghidour et al., 2021). This process results in token sequences that are shorter in length but deeper in hierarchy, effectively compressing data while maintaining high reconstruction fidelity. However, despite the advantages of RVQ in data compression, generative modeling on RVQ-based token sequences introduces new challenges. The hierarchical depth of these token sequences complicates the modeling process, particularly for autoregressive models whose sampling steps typically scale with the product of sequence length and depth. (Lee et al., 2022). Although non-autoregressive approaches have been explored along either sequence length or depth (Borsos et al., 2023; Copet et al., 2024; Kim et al., 2024a), existing methods do not effectively eliminate the sampling complexity associated with both dimensions simultaneously.

In this paper, we present ResGEN, an efficient RVQ-based generative modeling designed to achieve high-fidelity sample quality *without compromising sampling speed*. Our key innovation lies in the direct prediction of vector embeddings of collective tokens rather than predicting each token individually. By forecasting cumulative embeddings, we can estimate correlated tokens across different depths, aligning naturally with the RVQ quantization process. Additionally, we extend our approach involving a token masking strategy and a multi-token prediction mechanism within a principled probabilistic framework using a discrete diffusion process and variational inference. This approach allows us to decouple sampling complexity from both sequence length and depth, resulting in a model that generates high-fidelity samples efficiently.

We validate the efficacy and generalizability of ResGEN across two real-world generative tasks: conditional image generation on ImageNet 256×256 and zero-shot text-to-speech synthesis. Experimental results demonstrate superior performance over autoregressive counterparts in these tasks. Furthermore, as we scale the depth of RVQ, ResGEN exhibits enhanced sampling quality or faster speeds compared to similar-sized baseline generative models. We also analyze model characteristics exhibited with different RVQ depths and sampling steps in our ablation study.

The rest of the paper is organized as follows. In Section 3, we introduce the ResGEN framework, detailing the formulation of masked token prediction as a discrete diffusion process and the decoupling of generation iteration from token sequence length and depth. We also compare our approach with previous methods, highlighting the advantages of our strategy. In Section 5, we present experimental results that validate the performance of ResGEN, along with an ablation study on model performance with different RVQ depths and sampling steps. Finally, in Section 6, we discuss potential applications and future directions of our work.

## 2 Background

**Masked Token Modeling.** Masked token modeling, introduced in prior work (Chang et al., 2022), is a generative framework that operates on token sequences derived from the quantized encoder outputs of a Vector Quantized Variational AutoEncoder (VQ-VAE) (Van Den Oord et al., 2017). The core idea involves randomly masking a subset of input tokens and training the model to predict these masked tokens using a cross-entropy loss.

Formally, given a token sequence $\boldsymbol{x} \in \mathbb{N}^L$ and a corresponding binary mask $\boldsymbol{m} \in \{0, 1\}^L$, where each $\boldsymbol{m}_i = 0$ indicates that token $\boldsymbol{x}_i$ is masked, we create a masked token sequence $\boldsymbol{x} \odot \boldsymbol{m}$ by element-wise multiplying $\boldsymbol{x}$ and $\boldsymbol{m}$. The training objective is then formulated as:

$$\mathcal{L}_{\text{mask}}(\boldsymbol{x}, \boldsymbol{m}; \theta) = -\sum_{\substack{i \in [1, L], \\ \boldsymbol{m}_i = 0}} \log p_\theta(\boldsymbol{x}_i | \boldsymbol{x} \odot \boldsymbol{m}), \tag{1}$$

where $\theta$ denotes the model parameters. The masking process involves selecting a number of tokens $n$ to mask, determined by a masking schedule $n = \lceil \gamma(r) \cdot L \rceil$. Here, $r$ indicates the current time step in the unmasking process, ranging from zero to one, and $\gamma(\cdot)$ is a pre-defined masking scheduling function that monotonically decreases from one to zero as $r$ increases. During training, $r$ is sampled from a uniform distribution.

In the decoding phase, the model employs an iterative prediction process to progressively fill in the masked sequence. At each iteration, the masking ratio $r$ is updated to linearly increase from zero to one. Starting with an entirely masked token sequence, the model predicts the masked tokens, and a subset of these predicted tokens is selected to be unmasked based on confidence scores calculated through prediction probabilities. The number of tokens to unmask at each iteration is determined by the masking schedule.

**Residual Vector Quantization.** Residual Vector Quantization (RVQ) has been proposed to improve VQ-VAEs. While previous VQ-VAEs quantize an input by replacing each encoded vector with the nearest embedding from a codebook, RVQ iteratively applies vector quantization to the residuals of previous quantizations.

Formally, let the output of the encoder in a VQ-VAE at the position $i$ be $\boldsymbol{h}_{i,0}$. The residual vector quantizer maps it to a sequence of quantized tokens $\boldsymbol{x} \in \mathbb{N}^{L \times D}$, where $D$ is the total depth of the

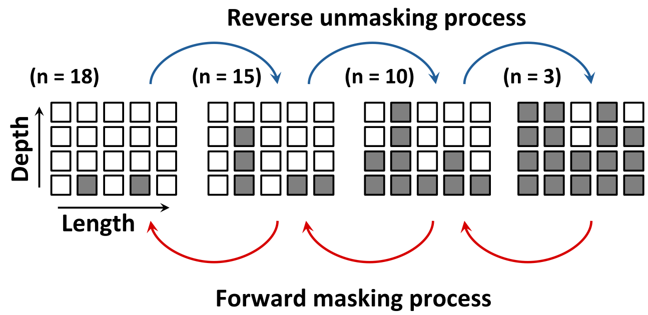


**Figure 1:** Overview of the Forward Masking and Reverse Unmasking Process. The forward masking process (depicted in red) moves from right to left, progressively increasing the number of masked tokens, while the reverse unmasking process (depicted in blue) moves from left to right, gradually revealing tokens. White boxes represent masked tokens, and colored boxes represent unmasked tokens. Our method iteratively predicts the masked tokens and replaces them with the predicted values, reducing the number of masked tokens at each step.

RVQ process:

$$\boldsymbol{x}_{i,j} = \underset{v \in \{1,\dots,V\}}{\arg\min} \, \|\boldsymbol{h}_{i,j-1} - \boldsymbol{e}(v;j)\|^2, \quad \boldsymbol{h}_{i,j} = \boldsymbol{h}_{i,j-1} - \boldsymbol{e}(\boldsymbol{x}_{i,j};j) \quad \text{for all } j \in [1, D], \tag{2}$$

where $\boldsymbol{e}(v;j)$ is the $v$-th vector embedding from the codebook at depth $j$, and $V$ is the number of embeddings per depth. Here, $\boldsymbol{x}_{i,j}$ represents the selected embedding index for the $i$-th token at depth $j$, and $\boldsymbol{h}_{i,j}$ denotes the residual vector after the $j$-th quantization step.

The final reconstructed vector is obtained by summing the embeddings across all depths, $\boldsymbol{z}_i = \sum_{j=1}^{D} \boldsymbol{e}(\boldsymbol{x}_{i,j};j)$. This iterative quantization process enables RVQ to produce a quantized output that closely approximates the original encoder output by increasing the depth $D$ of quantization steps. As a result, RVQ effectively captures the most significant features in the lower quantization layers, while finer details are progressively captured in higher layers.

## 3 Method

In this section, we introduce our method, ResGEN, which iteratively fills tokens in a coarse-to-fine manner to achieve efficient and high-fidelity generative modeling with Residual Vector Quantization (RVQ). We structure our discussion into three main parts:

- We present a token masking strategy tailored for RVQ tokens and describe how we model masked token prediction by predicting sum of residual vector embeddings to decouple the generation iterations from the length and depth of token sequences.
- We show that our proposed token masking and multi-token prediction method can be formulated within a principled probabilistic framework using a discrete diffusion process and variational inference.
- We detail the training and sampling techniques of ResGEN, focusing on the implementation of the mixture of Gaussians for latent embedding estimation and enhanced sampling strategies based on model confidence scores.

### 3.1 Masking and Prediction Task Design for RVQ Tokens

**Token Masking for RVQ Tokens.** Our masking strategy progressively masks tokens starting from the highest quantization layers, capitalizing on the hierarchical nature of RVQ where tokens at greater depths capture finer details.

Given a token sequence from RVQ, $\boldsymbol{x} \in \mathbb{N}^{L \times D}$, with sequence length $L$ and depth $D$, we apply a binary mask $\boldsymbol{m} \in \{0,1\}^{L \times D}$, where each $\boldsymbol{m}_{i,j}$ indicates whether the token $\boldsymbol{x}_{i,j}$ is masked ($\boldsymbol{m}_{i,j} = 0$) or not ($\boldsymbol{m}_{i,j} = 1$). The total number of tokens to mask is determined by a masking schedule,

$n = \lceil \gamma(r) \cdot L \cdot D \rceil$. Here, $r$ indicates the current time step in the unmasking process, ranging from zero to one, and $\gamma(\cdot)$ is a pre-defined masking scheduling function that monotonically decreases from one to zero as $r$ increases. During training, $r$ is sampled from a uniform distribution.

To distribute the $n$ masked tokens across the $L$ positions, the number of tokens to mask at each position $i$, denoted by $k_i$, is sampled without replacement from a multinomial distribution with equal probability across all positions, ensuring that $\sum_{i=1}^{L} k_i = n$. At each position $i$, $k_i$ tokens are masked starting from the highest depth $j = D$ and moving towards lower depths. This ensures that finer details captured at higher depths are masked before coarser information at lower depths, as illustrated in Figure 1.

**Multi-Token Prediction of Masked Tokens.** We describe the training and decoding phases of our multi-token prediction strategy, which efficiently predicts masked tokens by focusing on predicting the aggregated vector embeddings $z$ of collective tokens rather than the individual tokens $x$.

TRAINING: Given the input sequence $\boldsymbol{x}$ and the corresponding mask $\boldsymbol{m}$, the model predicts the sum of masked embeddings $\boldsymbol{z}$ such that $\boldsymbol{z}_i = \sum_j \boldsymbol{e}(\boldsymbol{x}_{i,j}; j) \odot (1 - \boldsymbol{m}_{i,j})$ rather than the target tokens directly, where $\boldsymbol{e}(v; j)$ denotes the $v$-th vector embedding from the RVQ codebook at depth $j$. The training objective is to maximize the log-likelihood of the sum of masked embeddings:

$$\mathcal{L}_{\text{mask}}(\boldsymbol{x}, \boldsymbol{m}; \theta) = - \sum_{\substack{i \in [1, L], \\ \sum_j \boldsymbol{m}_{i,j} < D}} \log p_\theta(\boldsymbol{z}_i | \boldsymbol{x} \odot \boldsymbol{m}), \tag{3}$$

where $\theta$ represents the model parameters and the summation over $i$ includes only those positions where at least one token is masked, denoted by $\sum_j \boldsymbol{m}_{i,j} < D$. To model the distribution $p_\theta$, we employ a mixture of Gaussian distributions. We modify the training objective to encourage the mixture component usage of the mixture of Gaussian distributions, which is described in Section 3.3.

This method avoids imposing conditional independence of tokens along the depth, which could harm model performance. Instead, it relies on the key idea that accurately predicting the vector embedding $\boldsymbol{z}_i$ is more critical than predicting the individual tokens $\boldsymbol{x}_i$, as the decoder of a VQ-VAE operates on vector embeddings.

SAMPLING: In the decoding phase, the model employs an iterative prediction process to progressively fill in the masked sequence. At each iteration, the masking ratio $r$ is updated to linearly increase from zero to one. Starting with an entirely masked token sequence, the model progressively fills in the sequence in a coarse-to-fine manner. At each step, the model predicts the cumulative masked token embedding $\boldsymbol{z}_i$. These predicted vectors are then quantized into tokens via RVQ quantization. A subset of these predicted tokens is randomly selected to be unmasked, where the number of tokens to unmask at each step is determined by the masking schedule. Although the quantization step at each sampling iteration involves sequential operations to reconstruct tokens from embeddings, it adds negligible overhead compared to the model forward pass.

We summarize the training and sampling algorithms for ResGEN in Algorithm 1 and Algorithm 2, respectively, in Appendix.

### 3.2 FORMULATION WITHIN A PROBABILISTIC FRAMEWORK

We explain our masked token prediction method within a principled probabilistic framework using a discrete diffusion process and variational inference. This formulation allows us to understand the generation process as a likelihood-based model, providing a theoretical foundation for our approach.

**Forward Discrete Diffusion Process.** We can interpret the token masking process described in Section 3.1 as the forward process of a discrete diffusion model. In this forward diffusion process, tokens are progressively masked starting from the highest depth to the lowest. At each step $t$, the masking involves sampling the number of tokens to mask from a multivariate hypergeometric distribution, which is equivalent to sampling from a multinomial distribution without replacement.

The forward process is defined as:

$$q(\boldsymbol{x}^{(t+1)} \mid \boldsymbol{x}^{(t)}) = \frac{\prod_{i=1}^{L} \binom{D-\sum_{\tau=1}^{t} k_i^{(\tau)}}{k_i^{(t+1)}}}{\binom{LD-\sum_{\tau=1}^{t} n^{(\tau)}}{n^{(t+1)}}}, \quad \text{where} \quad \boldsymbol{x}_{i,j}^{(t+1)} = \begin{cases} \boldsymbol{x}_{i,j}^{(t)} & \text{if } j \leq D - \sum_{\tau=1}^{t} k_i^{(\tau)} \\ \phi & \text{otherwise} \end{cases},$$

where $\phi$ denotes the masked token. This sequential sampling without replacement allows for direct sampling of any $\boldsymbol{x}^{(t)}$ from $\boldsymbol{x}^{(0)}$ and provides closed-form expressions for the forward process marginals:

$$q(\boldsymbol{x}^{(t)} \mid \boldsymbol{x}^{(0)}) = \frac{\prod_{i=1}^{L} \binom{D}{\sum_{\tau=1}^{t} k_i^{(\tau)}}}{\binom{LD}{\sum_{\tau=1}^{t} n^{(\tau)}}} \quad \text{and} \quad q(\boldsymbol{x}^{(t)} \mid \boldsymbol{x}^{(t+1)}, \boldsymbol{x}^{(0)}) = \frac{\prod_{i=1}^{L} \binom{\sum_{\tau=1}^{t+1} k_i^{(\tau)}}{k_i^{(t+1)}}}{\binom{\sum_{\tau=1}^{t+1} n^{(\tau)}}{n^{(t+1)}}},$$

$$\text{where } \boldsymbol{x}_{i,j}^{(t)} = \begin{cases} \boldsymbol{x}_{i,j}^{(0)} & \text{if } j \leq D - \sum_{\tau=1}^{t} k_i^{(\tau)} \\ \phi & \text{otherwise} \end{cases}.$$

**Reverse Discrete Diffusion Process.** In the reverse process, we aim to recover the original tokens from the masked sequences. Given $\boldsymbol{x}^{(t+1)}$, we predict $\boldsymbol{x}^{(0)}$ by sampling from $p_\theta(\boldsymbol{x}^{(0)} \mid \boldsymbol{x}^{(t+1)})$. The reverse process is formulated as:

$$p_\theta(\boldsymbol{x}^{(t)} \mid \boldsymbol{x}^{(t+1)}) = \sum_{\boldsymbol{x}^{(0)}} q(\boldsymbol{x}^{(t)} \mid \boldsymbol{x}^{(t+1)}, \boldsymbol{x}^{(0)}) p_\theta(\boldsymbol{x}^{(0)} \mid \boldsymbol{x}^{(t+1)}). \tag{4}$$

This formulation allows us to compute the variational lower bound of the data log-likelihood:

$$\mathbb{E}_q \left[ \underbrace{D_{\mathrm{KL}}\Big(q(\boldsymbol{x}^{(T)} \mid \boldsymbol{x}^{(0)}) \parallel p(\boldsymbol{x}^{(T)})\Big)}_{\mathcal{L}_T} + \sum_{t \geq 1} \underbrace{D_{\mathrm{KL}}\Big(q(\boldsymbol{x}^{(t)} \mid \boldsymbol{x}^{(t+1)}, \boldsymbol{x}^{(0)}) \parallel p_\theta(\boldsymbol{x}^{(t)} \mid \boldsymbol{x}^{(t+1)})\Big)}_{\mathcal{L}_t} - \mathcal{L}_0 \right].$$

Here, $\mathcal{L}_T$ is the prior loss, which becomes zero since $\boldsymbol{x}^{(T)}$ is fully masked, $\mathcal{L}_t$ are the diffusion losses at each step $t$, and $\mathcal{L}_0 := \log p_\theta(\boldsymbol{x}^{(0)} \mid \boldsymbol{x}^{(1)})$ is the reconstruction loss. By combining the diffusion losses and the reconstruction loss, we can derive a simplified loss function:

$$\mathcal{L}_{\text{diffusion}}(\boldsymbol{x}^{(0)}; \theta) = \sum_{t \geq 1} - \log p_\theta(\boldsymbol{x}^{(0)} \mid \boldsymbol{x}^{(t)}). \tag{5}$$

This loss function weights each term equally, focusing on predicting the original tokens from the partially masked sequences at each step.

**Latent Modeling with Variational Inference.** To enhance efficiency and capture dependencies across token depths, we adapt a multi-token prediction method inspired by CLaM-TTS (Kim et al., 2024a). Instead of predicting tokens individually, we predict the cumulative vector embeddings representing the tokens across depths. This approach aligns naturally with the RVQ dequantization process and decouples the generation time complexity from the token depth.

The key idea is that accurately predicting the vector embedding $\boldsymbol{z}$ is more critical than predicting the individual tokens $\boldsymbol{x}^{(0)}$, as the decoder of a VQ-VAE operates on vector embeddings. Using variational inference, we establish an upper bound on the negative log-likelihood:

$$- \log p_\theta(\boldsymbol{x}^{(0)} \mid \boldsymbol{x}^{(t)}) \leq \mathbb{E}_{q_{\boldsymbol{z}}} \left[ - \log p(\boldsymbol{x}^{(0)} | \boldsymbol{z}, \boldsymbol{x}^{(t)}) - \log \frac{p_\theta(\boldsymbol{z} \mid \boldsymbol{x}^{(t)})}{q(\boldsymbol{z} \mid \boldsymbol{x}^{(0)}, \boldsymbol{x}^{(t)})} \right].$$

By assuming that $p(\boldsymbol{x}^{(0)} | \boldsymbol{z}, \boldsymbol{x}^{(t)})$ corresponds to the RVQ quantization and $q(\boldsymbol{z} \mid \boldsymbol{x}^{(0)})$ corresponds to the RVQ dequantization of the masked tokens, we can focus on the remaining terms that have non-negligible gradients:

$$\mathcal{L}_{\text{simple}}(\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(t)}; \theta) = - \log p_\theta(\boldsymbol{z} | \boldsymbol{x}^{(t)}), \tag{6}$$

which is equivalent to the prediction loss in Equation 3.

### 3.3 TRAINING AND SAMPLING TECHNIQUES

**Mixture of Gaussians Implementation.** Our model utilizes a mixture of Gaussian distributions to represent the distribution over latent embeddings. Specifically, for each token position $i$, the model outputs the mixture probabilities $\boldsymbol{\pi}_i = \{\pi_i^{(\nu)}\}_{\nu=1}^K$, the mean vectors for each mixture component $\{\boldsymbol{\mu}_i^{(\nu)}\}_{\nu=1}^K$, and additional scale and shift parameters for affine transformations $a_i \in \mathbb{R}$ and $\boldsymbol{b}_i \in \mathbb{R}^H$, where $K$ is the number of mixture components and $H$ is the embedding dimension.

TRAINING OBJECTIVE MODIFICATION From Equation 3, the log-likelihood of the target embedding $\boldsymbol{z}_i$ is formulated as $\log p_\theta(\boldsymbol{z}_i|\boldsymbol{x} \odot \boldsymbol{m}) = -\log a_i + \log \sum_\nu \pi_i^{(\nu)} \mathcal{N}(\tilde{\boldsymbol{z}}_i; \boldsymbol{\mu}_i^{(\nu)}, \boldsymbol{I})$, where $\tilde{\boldsymbol{z}}_i = (\boldsymbol{z}_i - \boldsymbol{b}_i)/a_i$. To further encourage the usage of every mixture component, we modify the objective by decomposing it into a sum of classification and regression losses. Similar to prior work (Kim et al., 2024a), applying Jensen's inequality, we have:

$$
\begin{aligned}
&-\log a_i - \log \sum_\nu \pi_i^{(\nu)} \mathcal{N}(\tilde{\boldsymbol{z}}_i; \boldsymbol{\mu}_i^{(\nu)}, \boldsymbol{I}) \\
&\leq -\log a_i \underbrace{- \sum_\nu q(\nu \mid \tilde{\boldsymbol{z}}_i, \boldsymbol{\mu}_i) \log \mathcal{N}(\tilde{\boldsymbol{z}}_i; \boldsymbol{\mu}_i^{(\nu)}, \boldsymbol{I})}_{\text{regression loss}} + \underbrace{\mathrm{D_{KL}}(q(\nu \mid \tilde{\boldsymbol{z}}_i, \boldsymbol{\mu}_i) \parallel \boldsymbol{\pi}_i)}_{\text{classification loss}},
\end{aligned}
$$

where $q(\nu \mid \tilde{\boldsymbol{z}}_i, \boldsymbol{\mu}_i)$ is an auxiliary distribution defined as $q(\nu \mid \tilde{\boldsymbol{z}}_i, \boldsymbol{\mu}_i) \propto \mathcal{N}(\tilde{\boldsymbol{z}}_i; \boldsymbol{\mu}_i^{(\nu)}, \boldsymbol{I})$. This choice of $q$ ensures that mixture components with mean vectors closer to $\tilde{\boldsymbol{z}}_i$ have higher probabilities, while all components retain non-zero probabilities. Consequently, every mixture component contributes to the training process, promoting higher component usage and diversity in the model's predictions.

LOW-RANK PROJECTION Increasing the number of mixture components $K$ leads to a substantial growth in the output dimensionality of the model, as it scales with $K \times H$. To accommodate a high number of mixtures without incurring excessive computational costs, we adopt a low-rank projection approach following the methodology of the prior work (Kim et al., 2024a).

In this approach, the model outputs low-rank mean vectors $\{\tilde{\boldsymbol{\mu}}_i^{(\nu)}\}_{\nu=1}^K$, which are then transformed using trainable parameters $\boldsymbol{M}^{(\nu)}$ and $\boldsymbol{s}^{(\nu)}$: $\boldsymbol{\mu}_i^{(\nu)} = \boldsymbol{M}^{(\nu)} \tilde{\boldsymbol{\mu}}_i^{(\nu)} + \boldsymbol{s}^{(\nu)}$. This decomposition allows for efficient computation of the squared distance $\|\tilde{\boldsymbol{z}}_i - \boldsymbol{\mu}_i^{(\nu)}\|^2$ by expanding it as follows:

$$
\begin{aligned}
&\|\tilde{\boldsymbol{z}}_i - \boldsymbol{\mu}_i\|^2 = \|\tilde{\boldsymbol{z}}_i - (\boldsymbol{M}\tilde{\boldsymbol{\mu}}_i + \boldsymbol{s})\|^2 \\
&= \tilde{\boldsymbol{z}}_i^T \tilde{\boldsymbol{z}}_i + \tilde{\boldsymbol{\mu}}_i^T (\boldsymbol{M}^T \boldsymbol{M}) \tilde{\boldsymbol{\mu}}_i + \boldsymbol{s}^T \boldsymbol{s} - 2(\boldsymbol{M}^T \tilde{\boldsymbol{z}}_i)^T \tilde{\boldsymbol{\mu}}_i - 2\tilde{\boldsymbol{z}}_i^T \boldsymbol{s} + 2\tilde{\boldsymbol{\mu}}_i^T \boldsymbol{M}^T \boldsymbol{s},
\end{aligned} \tag{7}
$$

where we omit $\nu$ for simplicity. This low-rank projection enables the model to handle a large number of mixture components without significant overhead, thereby enhancing both the scalability and performance of the generative process.

**Enhanced Sampling with Confidence Scores.** To further improve the sampling process, we incorporate prediction probabilities of the model. Inspired by MaskGIT (Chang et al., 2022) and GIVT (Tschannen et al., 2023), we unmask tokens based on the predictive probabilities provided by the model. Specifically, we use the log probability derived from the mixture of Gaussian distributions as confidence scores for all masked tokens at each position $i$. Tokens with higher confidence scores are more likely to be unmasked and filled in earlier steps of the iterative generation process.

## 4 RELATED WORK

Vector-quantized (VQ) token-based generative models have emerged to harness the powerful generative capabilities of transformers for both autoregressive and non-autoregressive modeling. VQ-GAN (Esser et al., 2021) and DALL-E (Ramesh et al., 2021) leverage these discrete representations for image synthesis using transformers, facilitating high-quality generation with manageable computational resources.

Discrete diffusion models have been proposed to model token sequences by iteratively refining corrupted tokens or progressively unmasking masked tokens (Austin et al., 2021; Chang et al., 2022; Gu et al., 2022). MaskGIT (Chang et al., 2022) and VQ-Diffusion (Gu et al., 2022) focus on masked token prediction for flat token sequences, improving sampling efficiency over autoregressive models. GIVT (Tschannen et al., 2023) introduces a method that replaces softmax-based token prediction with mixture-of-Gaussians-based vector prediction in masked token prediction, progressively filling masked positions with predicted vectors.

However, these methods primarily deal with flat token sequences and do not consider the hierarchical depth inherent in RVQ. RQ-Transformer (Lee et al., 2022) was the first to demonstrate generative modeling on RVQ tokens sing an autoregressive model over the product of sequence length and depth, resulting in increased computational complexity. CLaM-TTS employs vector prediction for multi-token prediction but operates in an autoregressive manner along the sequence length. Vall-E (Wang et al., 2023) predicts the tokens at the first depth autoregressively and then predicts the remaining tokens at each depth in a single forward pass sequentially. SoundStorm (Borsos et al., 2023) generates tokens using masked token prediction given semantic tokens but still has sampling time complexity that increases linearly with the residual quantization depth. NaturalSpeech 2 (Shen et al., 2023) employs diffusion-based generative modeling on the RVQ embedding space.

In contrast to these approaches, our method offers a more efficient solution for generative modeling with RVQ tokens. We propose a strategy that predicts the vector embedding of masked tokens, effectively decoupling the sampling time complexity from both sequence length and token depth. By focusing on predicting cumulative vector embeddings rather than individual tokens, our method efficiently handles the hierarchical structure of tokens, offering enhanced sampling efficiency and high-fidelity generation.

# 5 Experiments

In this section, we demonstrate the superior performance of our approach in both image generation and text-to-speech synthesis, highlighting its quality and efficiency. In the first subsection, we evaluate ResGEN for class-conditional image generation on ImageNet (Krizhevsky et al., 2017) at a resolution of $256 \times 256$. In the next subsection, we showcase the versatility of our framework by demonstrating its performance in text-to-speech synthesis, where it consistently generates high-quality 44kHz audio. In the last subsection, we present an ablation study on the results of the sampling algorithm under various schedules, showing that our method remains robust even when the number of time steps is reduced.

We train our method based on a similar architecture to DiT (Peebles & Xie, 2023), adopting the XLarge version but replacing the linear layers with adaptive layer normalization layers conditioned on bias parameters. For the ImageNet 256x256 task, all variants of ResGEN are trained with a batch size of 256 across 4 GPUs for 2.75M to 4M iterations. To increase the depth of the RVQ, we warm-start from the checkpoint of RQ-VAE (Lee et al., 2022), excluding the attention layers, and reduce the latent dimension from 256 to 64. These models are trained for an additional 1M steps each, with and without adversarial training, following the same configuration as prior work.

For the Text-to-Speech task, our model is trained using the same configuration as in prior work (Kim et al., 2024a), utilizing 2 GPUs for 275M iterations. We employ 4 transformer layers to train a linear regression duration predictor for the text inputs, built on top of the pretrained text encoder.

## 5.1 Experimental Setting

**Experiment Tasks** To assess the effectiveness of our method, we selected representative tasks from each domain. For the vision domain, we focused on conditional image generation tasks. In the audio domain, we evaluated our model using two tasks inspired by Voicebox (Le et al., 2023): 1) *continuation*: given a text and a 3-second segment of ground truth speech, the goal is to generate seamless speech that continues in the same style as the provided segment; 2) *cross-sentence*: given a text, a 3-second speech segment, and its transcript (which differs from the text), the objective is to generate speech that reads the text in the style of the provided segment.

**Table 1:** Comparison of generation quality between the RQ-transformer and ResGen using the same RVQ tokens.

| Model | Params ↓ | FID ↓ | Inference Time(s) ↓ |
|---|---|---|---|
| RQ-transformer (Lee et al., 2022) | 821M | 13.11 | 2.38s |
| ResGEN | 594M | 13.07 | 1.38s |

**Table 2:** Comparison of various generative models on class-conditional ImageNet at a resolution of 256×256. Inference time is calculated relative to ResGen. Performance with and without CFG is measured using the same number of steps. Models marked with * are sourced from the original papers.

| Model | Params ↓ | FID (w/o CFG) ↓ | FID (w/ CFG) ↓ | Inference Time ↓ |
|---|---|---|---|---|
| MaskGiT | 277M | 6.18* | - | 2.0 |
| RQ-Transformer | 821M | 13.11* | - | 21.0 |
| DiT | 675M | 9.62* | 2.27* | 45.0 |
| VAR-d20 | 600M | 8.51 | 2.57* | 0.5 |
| ResGen | 574M | 7.84 | 2.75 | 2.0 |

**Evaluation Metrics** For vision tasks, we employ the Fréchet Inception Distance (FID) (Heusel et al., 2017) for comparing it with other state-of-the-art image generative models. For a fair comparison, we follow the evaluation procedure presented in (Lee et al., 2022). For audio tasks, we evaluate the models using the following objective metrics: Character Error Rate (CER), Word Error Rate (WER), and Speaker Similarity (SIM), as described in VALL-E (Wang et al., 2023) and CLaM-TTS (Kim et al., 2024b). CER and WER measure the model's intelligibility and robustness, while SIM assesses how accurately the model captures the speaker's identity.

**Baselines** In the vision domain, we compare our models with recent generative model families, including (1) *autoregressive models*: RQ-transformer (Lee et al., 2022), VAR (Tian et al., 2024); and (2) *non-autoregressive models*: MaskGiT (Chang et al., 2022), DiT (Peebles & Xie, 2023). For the audio task, we benchmark the proposed model against state-of-the-art TTS models, including (1) *autoregressive models*: VALL-E (Wang et al., 2023), SPEAR-TTS (Kharitonov et al., 2023), and CLaM-TTS (Kim et al., 2024b); and (2) *non-autoregressive models*: YourTTS (Casanova et al., 2022), VoiceBox (Le et al., 2023), and DiTTo-TTS (Lee et al., 2024).

### 5.2 Effectiveness of our generative modeling

In our experiments, we compare our method with autoregressive models that generate RVQ tokens. In the Text-to-Speech experiments, we train our method using an RVQ-VAE similar to MelVAE from CLaM-TTS. As shown in Table 3 and Table 4, our method outperforms the baselines across all metrics, achieving lower error rates, higher speaker similarity scores, and requiring fewer inference steps. These results demonstrate that our method effectively generates RVQ tokens with a small number of iterations. Notably, our method uses only 16 iterations, which is fewer than the RVQ depth of 32. Although our results are not on par with the state-of-the-art method, DiTTo-TTS, our approach achieves the smallest number of sampling iterations among the baselines, highlighting its efficiency in terms of computational complexity.

For the image conditional generation task, we evaluate our method using the RQ-transformer with the same number of RVQ tokens. As shown in 1, our method not only outperforms the baseline but also achieves faster inference times. Notably, our model is trained with fewer parameters, totaling only 574M.

**Table 3:** Performances for the English-only *continuation* task. The boldface indicates the best result, the underline denotes the second best, and the asterisk denotes the score reported in the baseline paper. The inference time indicates the generation time of 10s speech.

| | Objective Metrics | | | | |
|---|---|---|---|---|---|
| **Model** | **WER ↓** | **CER ↓** | **SIM-o ↑** | **SIM-r ↑** | **Inference Steps ↓** |
| Ground Truth | 2.2* | 0.61* | 0.754* | 0.754* | n/a |
| YourTTS (Casanova et al., 2022) | 7.57 | 3.06 | 0.3928 | - | **1** |
| Vall-E (Wang et al., 2023) | 3.8* | - | 0.452* | 0.508* | - |
| Voicebox (Le et al., 2023) | <u>2.0*</u> | - | **0.593*** | **0.616*** | 64 |
| CLaM-TTS (Kim et al., 2024a) | 2.36* | 0.79* | 0.4767* | 0.5128* | - |
| DiTTo-en-L (Lee et al., 2024) | **1.85*** | **0.50*** | <u>0.5596*</u> | <u>0.5913*</u> | 25 |
| ResGEN | <u>1.99</u> | <u>0.55</u> | 0.5341 | 0.5627 | <u>16</u> |

**Table 4:** Performances for the English-only *cross-sentence* task.

| **Model** | **WER ↓** | **CER ↓** | **SIM-o ↑** | **SIM-r ↑** |
|---|---|---|---|---|
| YourTTS (Casanova et al., 2022) | 7.92 (7.7*) | 3.18 | 0.3755 (0.337*) | - |
| Vall-E (Wang et al., 2023) | 5.9* | - | - | 0.580* |
| SPEAR-TTS (Kharitonov et al., 2023) | - | 1.92* | - | 0.560* |
| Voicebox (Le et al., 2023) | <u>1.9*</u> | - | **0.662*** | **0.681*** |
| CLaM-TTS (Kim et al., 2024a) | 5.11* | 2.87* | 0.4951* | 0.5382* |
| DiTTo-en-L (Lee et al., 2024) | 2.69* | <u>0.91*</u> | <u>0.6050*</u> | <u>0.6355*</u> |
| ResGEN | **1.83** | **0.50** | 0.5562 | 0.6073 |

### 5.3 Comparison with other methods

In the vision domain, we demonstrated the superiority of our method by comparing it with other approaches. As shown in 2, our method not only achieves faster generation speed but also demonstrates superior generation quality compared to other models with similar parameter sizes.

## 6 Conclusion

In this work, we propose ResGEN, an efficient RVQ-based discrete diffusion model that generates high-fidelity samples while maintaining fast sampling speeds. By directly predicting the vector embedding of collective tokens, our method addresses the typical trade-offs between token depth and inference speed in vector-quantized generative models. We further demonstrate the effectiveness of token masking and multi-token prediction within a principled probabilistic framework, employing a discrete diffusion process and variational inference. Our experiments on both conditional image generation and zero-shot text-to-speech synthesis validate the strong performance of ResGEN, which performs comparably to or exceeds autoregressive models in terms of fidelity and sampling speed. As we scale the depth of RVQ, our model exhibits improvements in generation fidelity or efficiency, showing its scalability and generalizability across different modalities.

## References

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

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.

Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Yuanzhen Li, Tomer Michaeli, et al. Lumiere: A space-time diffusion model for video generation. *arXiv preprint arXiv:2401.12945*, 2024.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023.

Zalán Borsos, Matt Sharifi, Damien Vincent, Eugene Kharitonov, Neil Zeghidour, and Marco Tagliasacchi. Soundstorm: Efficient parallel audio generation. *arXiv preprint arXiv:2305.09636*, 2023.

Edresson Casanova, Julian Weber, Christopher D Shulby, Arnaldo Candido Junior, Eren Gölge, and Moacir A Ponti. Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone. In *International Conference on Machine Learning (ICML)*, pp. 2709–2720. PMLR, 2022.

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

Yongjian Chen, Tao Guan, and Cheng Wang. Approximate nearest neighbor search by residual vector quantization. *Sensors*, 10(12):11259–11273, 2010.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. *Advances in Neural Information Processing Systems*, 36, 2024.

Zheng Ding, Mengqi Zhang, Jiajun Wu, and Zhuowen Tu. Patched denoising diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2023.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

Zach Evans, CJ Carr, Josiah Taylor, Scott H Hawley, and Jordi Pons. Fast timing-conditioned latent audio diffusion. *arXiv preprint arXiv:2402.04825*, 2024.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10696–10706, 2022.

Yingqing He, Shaoshu Yang, Haoxin Chen, Xiaodong Cun, Menghan Xia, Yong Zhang, Xintao Wang, Ran He, Qifeng Chen, and Ying Shan. Scalecrafter: Tuning-free higher-resolution visual generation with diffusion models. In *The Twelfth International Conference on Learning Representations*, 2023.

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.

Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10124–10134, 2023.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Eugene Kharitonov, Damien Vincent, Zalán Borsos, Raphaël Marinier, Sertan Girgin, Olivier Pietquin, Matt Sharifi, Marco Tagliasacchi, and Neil Zeghidour. Speak, Read and Prompt: High-Fidelity Text-to-Speech with Minimal Supervision. *Transactions of the Association for Computational Linguistics*, 11:1703–1718, 12 2023. ISSN 2307-387X. doi: 10.1162/tacl_a_00618.

Jaehyeon Kim, Keon Lee, Seungjun Chung, and Jaewoong Cho. Clam-tts: Improving neural codec language model for zero-shot text-to-speech. In *The Twelfth International Conference on Learning Representations*, 2024a.

Jaehyeon Kim, Keon Lee, Seungjun Chung, and Jaewoong Cho. CLam-TTS: Improving neural codec language model for zero-shot text-to-speech. In *International Conference on Learning Representations (ICLR)*, 2024b. URL `https://openreview.net/forum?id=ofzeypWosV`.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, and Wei-Ning Hsu. Voicebox: Text-guided multilingual universal speech generation at scale. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 14005–14034. Curran Associates, Inc., 2023.

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

Keon Lee, Dong Won Kim, Jaehyeon Kim, and Jaewoong Cho. Ditto-tts: Efficient and scalable zero-shot text-to-speech with diffusion transformer. *arXiv preprint arXiv:2406.11427*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pp. 8821–8831. Pmlr, 2021.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

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

Paul K Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quitry, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, et al. Audiopalm: A large language model that can speak and listen. *arXiv preprint arXiv:2306.12925*, 2023.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

Kai Shen, Zeqian Ju, Xu Tan, Eric Liu, Yichong Leng, Lei He, Tao Qin, Jiang Bian, et al. Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers. In *The Twelfth International Conference on Learning Representations*, 2023.

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

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

Michael Tschannen, Cian Eastwood, and Fabian Mentzer. Givt: Generative infinite-vocabulary transformers. *arXiv preprint arXiv:2312.02116*, 2023.

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

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2021.

# A Appendix

## A.1 Training and Sampling Algorithms

**Algorithm 1** Training

**procedure** BinaryMask($n, L, D$)
  Sample $k_{1:L}$ **without replacement** with total draws $n$.
  **for** $i = 1$ to $L$ **do**
    $m_{i,1:(D-k_i)} \leftarrow 1$
    $m_{i,(D-k_i+1):D} \leftarrow 0$
  **end for**
  **return** $m$
**end procedure**

**repeat**
  $x \sim p_{data}$
  $r \sim \text{Uniform}[0, 1)$
  $n \leftarrow \lceil \gamma(r) \cdot L \cdot D \rceil$
  $m \leftarrow \text{BinaryMask}(n, L, D)$
  $z \leftarrow \sum_j \left(e(x_{:,j}; j) \odot (1 - m_{:,j})\right)$
  Take a gradient descent step on:
    $-\nabla_\theta \log p_\theta(z|x \odot m)$
**until** converged

**Algorithm 2** Sampling

**procedure** BinaryUnmask($n, L, D, m$)
  Compute the number of masked tokens $q_i = \sum_{j=1}^{D}(1 - m_{i,j})$
  Sample $k_{1:L}$ from a multivariate hypergeometric distribution with maximum number of selection $q_i$, total draws $\sum_i q_i - n$.
  **for** $i = 1$ to $L$ **do**
    $m[i, (D - q_i + 1){:}(D - q_i + k_i)] \leftarrow 1$
  **end for**
  **return** $m$
**end procedure**

Initialize a fully masked sequence $x \in \mathbb{N}^{L \times D}$
Initialize mask $m \in \{0, 1\}^{L \times D}$ with zeros.
**for** $t = 1, \ldots, T$ **do**
  $z \sim p_\theta(z|x \odot m)$
  Apply residual vector quantization for masked tokens:
    $x \leftarrow RVQ(z, m)$
  $r \leftarrow \frac{t}{T}$
  $n \leftarrow \lceil \gamma(r) \times L \times D \rceil$
  $m \leftarrow \text{BinaryUnmask}(n, L, D, m)$
**end for**
**return** $x$