

Contrastive Deep Learning for Variant Detection in Wastewater Genomic Sequencing

Adele Chinda^{1,*}, Richmond Azumah^{1,*}, Hemanth Demakethepalli Venkateswara¹

¹Georgia State University, Atlanta, Georgia, USA

{achinda1, razumah1}@student.gsu.edu, hvenkateswara@gsu.edu

Abstract

Wastewater-based genomic surveillance has emerged as a powerful tool for population-level viral monitoring, offering comprehensive insights into circulating viral variants across entire communities. However, this approach faces significant computational challenges stemming from high sequencing noise, low viral coverage, fragmented reads, and the complete absence of labeled variant annotations. Traditional reference-based variant calling pipelines struggle with novel mutations and require extensive computational resources. We present a comprehensive framework for unsupervised viral variant detection using Vector-Quantized Variational Autoencoders (VQ-VAE) that learns discrete codebooks of genomic patterns from k-mer tokenized sequences without requiring reference genomes or variant labels. Our approach extends the base VQ-VAE architecture with masked reconstruction pretraining for robustness to missing data and contrastive learning for highly discriminative embeddings. Evaluated on SARS-CoV-2 wastewater sequencing data comprising approximately 100,000 reads, our VQ-VAE achieves 99.52% mean token-level accuracy and 56.33% exact sequence match rate while maintaining 19.73% codebook utilization (101 of 512 codes active), demonstrating efficient discrete representation learning. Contrastive fine-tuning with different projection dimensions yields substantial clustering improvements: 64-dimensional embeddings achieve +35% Silhouette score improvement ($0.31 \rightarrow 0.42$), while 128-dimensional embeddings achieve +42% improvement ($0.31 \rightarrow 0.44$), clearly demonstrating the impact of embedding dimensionality on variant discrimination capability. Our reference-free framework provides a scalable, interpretable approach to genomic surveillance with direct applications to public health monitoring.

1. Introduction

The COVID-19 pandemic has fundamentally demonstrated that rapid, comprehensive viral variant surveillance is critical for effective public health response and intervention strategies [1, 7]. Wastewater-based epidemiology (WBE) has emerged as a cost-effective, non-invasive approach to monitoring viral genetic diversity by analyzing viral genetic material shed by entire communities into municipal wastewater systems [2, 4]. Unlike traditional clinical testing approaches that are subject to testing availability, healthcare access disparities, and individual testing behavior, wastewater sampling provides unbiased, comprehensive snapshots of viral circulation patterns across populations, including contributions from asymptomatic and pre-symptomatic carriers who would otherwise go undetected [5]. This population-level surveillance capability makes wastewater monitoring particularly valuable for early warning systems and tracking the emergence of novel viral variants.

However, wastewater-based genomic sequencing presents unique and challenging computational problems that distinguish it from clinical sequencing. First, sequencing reads from wastewater are highly fragmented, typically ranging from 100-300 base pairs with significant technical and biological noise [3]. Second, viral RNA concentrations in wastewater are extremely low and heavily diluted by environmental material, human microbiome sequences, and other contaminants [1, 4]. Third, multiple viral strains and variants co-circulate simultaneously at varying population frequencies, creating complex mixtures that confound traditional variant calling approaches [6]. Traditional bioinformatics pipelines for variant identification rely heavily on alignment to reference genomes using tools such as BWA [8], followed by variant calling using methods like LoFreq [9] or iVar [7]. These reference-dependent approaches face critical limitations: they may fail to detect novel mutations that differ significantly from reference sequences, they struggle with low-frequency variants in mixed populations, and they require substantial com-

*Equal contribution.

putational resources (often hours per sample) that limit real-time surveillance capabilities.

Deep learning approaches offer significant promise for reference-free genomic modeling and analysis. Generative models have demonstrated success in creating realistic artificial human genomes [12], learning functional protein representations that capture mutational effects [13], and modeling complex DNA sequence patterns [23–25]. However, standard Variational Autoencoders (VAEs) with continuous latent spaces often produce overly smooth representations that are fundamentally unsuitable for modeling discrete mutational events in genomic sequences [15]. Furthermore, continuous VAEs frequently suffer from posterior collapse, where the latent codes become uninformative and the model degenerates into a simple autoregressive decoder [16].

Vector-Quantized Variational Autoencoders (VQ-VAE) [17] address these fundamental limitations through discrete latent representations, effectively preventing posterior collapse while enabling high-quality reconstruction and interpretable latent codes [18]. The discrete bottleneck enforces information compression into a finite codebook of learned patterns, naturally aligning with the discrete nature of genetic variation. Recent work has successfully applied VQ-VAE to DNA sequence modeling [22], demonstrating superior compression efficiency and variant-aware representation learning compared to continuous latent models. Additionally, self-supervised learning objectives including masked language modeling [27] and contrastive learning frameworks such as SimCLR [28] and MoCo [29] have fundamentally transformed representation learning across domains, with particularly strong transfer performance to protein sequences [25, 31] and other biological data [32].

Contributions. In this work, we present the first comprehensive application of discrete representation learning to wastewater genomic surveillance. Our specific contributions are: (1) A VQ-VAE architecture with Exponential Moving Average (EMA) quantization achieving 99.52% token-level reconstruction accuracy while maintaining 19.73% codebook utilization, demonstrating efficient discrete pattern learning; (2) Masked pretraining following BERT-style objectives that maintains approximately 95% reconstruction accuracy under 20% token corruption, enabling robust inference with missing or low-quality data; (3) Contrastive fine-tuning experiments with varying embedding dimensions (64-dim and 128-dim) showing +35% and +42% Silhouette score improvements respectively, clearly establishing the impact of representation capacity on variant discrimination; (4) Comprehensive evaluation demonstrating that discrete representation learning provides a scalable, reference-free alternative to traditional variant calling pipelines for wastewater surveillance applications.

2. Related Work

2.1. Wastewater Genomic Surveillance

Wastewater-based genomic surveillance has rapidly evolved as a critical tool for monitoring viral circulation at the population level. Crits-Christoph et al. [1] demonstrated that metagenomic sequencing of municipal sewage can effectively detect regionally prevalent SARS-CoV-2 variants, providing early warning signals for variant emergence and spread. Rothman et al. [3] extended this work by applying RNA viromics approaches to detect single-nucleotide variants in wastewater samples, revealing the fine-grained mutational landscape. However, these approaches fundamentally rely on reference genome alignment using tools such as BWA [8], followed by variant calling using sophisticated probabilistic methods like LoFreq [9] for low-frequency variant detection or iVar [7] for consensus sequence generation. These reference-dependent pipelines face several critical limitations: they may systematically miss novel variants that diverge significantly from reference sequences, they require extensive computational resources (typically 1-2 hours per sample), and they struggle with the complex mixture distributions present in wastewater where multiple variants co-circulate at varying frequencies.

2.2. Deep Generative Models for Genomics

Deep generative models have emerged as powerful tools for learning complex patterns in genomic data. Generative Adversarial Networks (GANs) have successfully generated realistic artificial human genomes that preserve population genetic structure and linkage disequilibrium patterns [12]. Variational Autoencoders have been applied to learn functional protein representations that accurately capture the effects of mutations on protein function [13], enabling zero-shot prediction of variant effects. More recently, transformer-based models such as DNABERT [23] and its successor DNABERT-2 [24] have achieved state-of-the-art performance on diverse genomic tasks including promoter prediction, splice site detection, and transcription factor binding site identification, leveraging masked language modeling pretraining on large genomic corpora. However, standard VAEs with continuous latent spaces frequently suffer from posterior collapse [15, 16], where the latent codes become uninformative and provide no meaningful representation of input variation. This fundamental limitation makes continuous VAEs poorly suited for modeling discrete mutational events and variant structures in genomic sequences.

2.3. Vector Quantization and Discrete Representations

Vector-Quantized Variational Autoencoders (VQ-VAE) [17] introduced discrete latent representations that

fundamentally avoid posterior collapse by replacing continuous latent distributions with discrete codebook lookups. The discrete bottleneck forces the model to learn a finite vocabulary of patterns that must be composed to reconstruct inputs, naturally encouraging meaningful, interpretable representations. VQ-VAE-2 [18] extended this framework hierarchically, learning multi-scale discrete representations that capture both fine-grained details and high-level structure, achieving state-of-the-art image generation quality. Masked VQ-VAE [19] improved computational efficiency through selective attention mechanisms that focus on informative regions. Most relevant to our work, VQ-DNA [22] first applied discrete latent representations to genomic sequence modeling, demonstrating superior compression efficiency and variant-aware representation learning compared to continuous alternatives. Their work established that discrete codebooks can effectively capture mutational patterns and variant structures.

2.4. Self-Supervised Learning in Biology

Self-supervised learning has fundamentally transformed representation learning by enabling models to learn from vast amounts of unlabeled data. Masked language modeling, introduced in BERT [27], trains models to predict randomly masked tokens from surrounding context, learning rich contextual representations. This paradigm has transferred exceptionally well to biological sequences [23, 25]. Contrastive learning frameworks including SimCLR [28] and MoCo [29] learn representations by maximizing agreement between differently augmented views of the same sample while minimizing agreement with other samples. These methods have been successfully applied to protein sequences [31], single-cell RNA sequencing data [32], and other biological modalities, consistently demonstrating that self-supervised pretraining substantially improves downstream task performance and produces more discriminative, biologically meaningful representations.

3. Method

Our framework consists of three complementary components working in concert: (1) quality-controlled k-mer tokenization that converts raw sequencing reads into fixed-length discrete token sequences, (2) a Vector-Quantized Variational Autoencoder with Exponential Moving Average updates and entropy regularization that learns a discrete codebook of genomic patterns, and (3) two self-supervised learning extensions—masked reconstruction pretraining for robustness to missing data and contrastive learning for discriminative embeddings. Figure ?? provides a comprehensive illustration of the complete pipeline from raw reads to discrete representations.

3.1. Data Preprocessing and K-mer Tokenization

Raw FASTQ sequencing reads from wastewater samples undergo rigorous quality control to remove low-quality bases and adapter contamination. We employ Trimmomatic [10] with the following parameters: leading and trailing base quality threshold of 3 (removing low-quality bases from read ends), sliding window quality filtering with window size 4 and average quality threshold 15 (ensuring consistent quality throughout reads), and minimum read length of 36 base pairs (discarding overly fragmented reads). Quality improvement is validated using FastQC [11] reports comparing pre- and post-filtering read quality distributions, GC content, sequence length distributions, and per-base quality scores.

Following quality control, we convert nucleotide sequences into discrete token sequences using k-mer extraction. For a DNA sequence $s = s_1s_2\dots s_n$ where each $s_i \in \{A, C, G, T\}$ represents a nucleotide base, we extract all overlapping k-mers with stride 1:

$$\text{tokens}(s) = [s_{1:k}, s_{2:k+1}, s_{3:k+2}, \dots, s_{n-k+1:n}] \quad (1)$$

Each k-mer $s_{i:i+k-1}$ is mapped to a unique integer identifier in vocabulary \mathcal{V} of size $V = 4^k + 1$, where 4^k accounts for all possible k-mer sequences and the additional token represents padding (PAD). We employ canonical k-mer representation where each k-mer and its reverse complement map to the same token identifier, reducing vocabulary size and improving biological interpretability. Based on empirical analysis of read lengths and computational efficiency, we select $k = 6$, yielding vocabulary size $V = 4,097$. All sequences are either padded with PAD tokens or truncated to a fixed maximum length $L = 150$ tokens, corresponding to the median read length in our dataset after quality filtering.

3.2. Vector-Quantized Variational Autoencoder

Our VQ-VAE architecture consists of three main components: an encoder that maps token sequences to continuous latent representations, a vector quantizer that discretizes these representations through learned codebook lookup, and a decoder that reconstructs token sequences from quantized latents. This architecture is illustrated in detail in Figure 1.

Encoder Architecture. The encoder E_ϕ maps input token sequences $x \in \mathbb{Z}^L$ (where \mathbb{Z}^L denotes L-length sequences of integer token IDs) to continuous latent representations $z_e \in \mathbb{R}^{L \times D}$:

$$z_e = E_\phi(x) \quad (2)$$

The encoder architecture consists of the following layers in sequence: (1) Token embedding layer mapping each token ID to a 128-dimensional continuous vector: $\text{Emb}: \mathbb{Z}^V \rightarrow \mathbb{R}^{128}$, producing embedding matrix $E \in \mathbb{R}^{L \times 128}$;

(2) Two 1-dimensional convolutional layers with kernel size 3, hidden dimension 256, and ReLU activations that capture local sequential dependencies and patterns in the k-mer sequences; (3) Layer normalization after each convolutional layer to stabilize training dynamics and accelerate convergence; (4) Dropout with probability $p = 0.1$ applied after layer normalization to prevent overfitting; (5) Final linear projection layer mapping from hidden dimension 256 to latent dimension $D = 64$. This encoder architecture effectively captures both local k-mer patterns through convolutional processing and produces compact 64-dimensional latent representations suitable for discrete quantization.

Vector Quantization. The vector quantizer discretizes continuous encoder outputs through nearest-neighbor lookup in a learned codebook $\mathcal{C} = \{e_1, e_2, \dots, e_K\}$ where each codebook vector $e_k \in \mathbb{R}^D$ and $K = 512$ is the total number of discrete codes. For each position i in the latent sequence, quantization proceeds as:

$$q_i = \arg \min_{k \in \{1, \dots, K\}} \|z_e^{(i)} - e_k\|_2^2 \quad (3)$$

$$z_q^{(i)} = e_{q_i} \quad (4)$$

where q_i is the index of the nearest codebook vector and $z_q^{(i)}$ is the quantized latent representation. The quantizer essentially converts each continuous latent vector $z_e^{(i)}$ into its nearest discrete code from the learned codebook, creating a sequence of discrete code indices.

Exponential Moving Average Updates. Rather than directly optimizing codebook vectors through backpropagation (which can lead to instability), we employ Exponential Moving Average (EMA) updates [17] to maintain and update the codebook. For each code k , we track the exponentially weighted count N_k of times the code is used and the exponentially weighted sum m_k of encoder outputs assigned to that code:

$$N_k^{(t)} = \gamma N_k^{(t-1)} + (1 - \gamma)n_k^{(t)} \quad (5)$$

$$m_k^{(t)} = \gamma m_k^{(t-1)} + (1 - \gamma) \sum_{i: q_i=k} z_e^{(i)} \quad (6)$$

where $\gamma = 0.95$ is the decay parameter, t indexes training iterations, and $n_k^{(t)}$ is the count of encoder outputs assigned to code k in the current batch. The codebook vector is then updated as:

$$e_k^{(t)} = \frac{m_k^{(t)}}{N_k^{(t)}} \quad (7)$$

This EMA update mechanism ensures that codebook vectors continuously adapt to track the distribution of encoder outputs without requiring explicit gradient-based optimization of the codebook parameters.

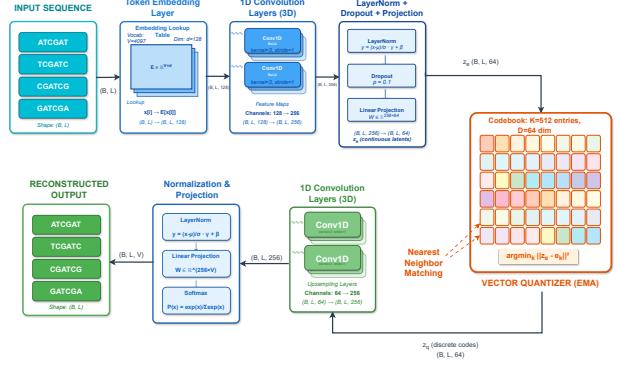


Figure 1. VQ-VAE Architecture Details. Encoder: token embedding ($4097 \rightarrow 128$) + 2 Conv1D layers (256 hidden, kernel=3) + LN + Dropout(0.1) $\rightarrow z_e$ ($L \times 64$). Quantizer: nearest-neighbor lookup in codebook (512 entries, 64-dim) with EMA updates ($=0.95$). Decoder: 2 Conv1D + LN + Linear projection to vocabulary logits. Red arrows show straight-through gradient flow.

Straight-Through Gradient Estimator. Since the quantization operation involves discrete argmin, it is non-differentiable and blocks gradient flow from decoder to encoder during backpropagation. We employ the straight-through estimator [33] which simply copies gradients from the quantized representation z_q directly to the encoder output z_e :

$$\nabla_{z_e} \mathcal{L} = \nabla_{z_q} \mathcal{L} \quad (8)$$

This gradient approximation, while biased, works effectively in practice and enables end-to-end training of the encoder-quantizer-decoder pipeline.

Decoder Architecture. The decoder D_θ reconstructs token logits from quantized latent representations $z_q \in \mathbb{R}^{L \times D}$:

$$\hat{x} = D_\theta(z_q) \in \mathbb{R}^{L \times V} \quad (9)$$

The decoder consists of: (1) Two 1-dimensional convolutional layers with kernel size 3 and hidden dimension 256, mirroring the encoder architecture; (2) Layer normalization after each convolutional layer; (3) Final linear projection layer mapping from hidden dimension 256 to vocabulary size $V = 4,097$, producing unnormalized logits $\hat{x}_{i,v}$ for each position i and vocabulary token v . The decoder architecture is intentionally symmetric to the encoder, facilitating effective reconstruction while maintaining computational efficiency.

Training Objective. The complete training objective combines three complementary loss terms:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}} + \beta \mathcal{L}_{\text{commit}} - \lambda H[\mathcal{C}] \quad (10)$$

The reconstruction loss $\mathcal{L}_{\text{recon}}$ is the cross-entropy between predicted and ground-truth tokens, computed only

over valid (non-padding) positions:

$$\mathcal{L}_{\text{recon}} = - \sum_{i \in \text{valid}} \log p_\theta(x_i | z_q^{(i)}) = - \sum_{i \in \text{valid}} \log \frac{\exp(\hat{x}_{i,x_i})}{\sum_{v=1}^V \exp(\hat{x}_{i,v})} \quad (11)$$

where $\hat{x}_{i,v}$ are the decoder logits for position i and vocabulary token v , and x_i is the ground-truth token at position i . This loss encourages accurate reconstruction of the input sequence from its quantized representation.

The commitment loss $\mathcal{L}_{\text{commit}}$ encourages encoder outputs to commit to codebook entries, preventing encoder outputs from growing arbitrarily large:

$$\mathcal{L}_{\text{commit}} = \frac{1}{L} \sum_{i=1}^L \|z_e^{(i)} - \text{sg}[z_q^{(i)}]\|_2^2 \quad (12)$$

where $\text{sg}[\cdot]$ denotes stop-gradient operation (treating the argument as constant during backpropagation). This loss applies gradients only to the encoder, pulling encoder outputs closer to their assigned codebook vectors. We use commitment weight $\beta = 0.1$.

The codebook entropy regularization term encourages diverse codebook utilization, preventing codebook collapse where only a few codes dominate:

$$H[\mathcal{C}] = - \sum_{k=1}^K p_k \log p_k \quad (13)$$

where p_k is the empirical frequency of code k being selected across all positions in the current training batch. Maximizing entropy (thus subtracting it in the loss with weight $\lambda = 3 \times 10^{-3}$) encourages uniform code utilization and prevents pathological collapse to a small subset of codes.

3.3. Masked Reconstruction Pretraining

Inspired by masked language modeling in BERT [27], we extend the base VQ-VAE with masked reconstruction pretraining to improve robustness to missing or corrupted data. During masked pretraining, we randomly select a fraction $p_{\text{mask}} = 0.2$ (20%) of non-padding token positions and replace them with a special MASK token. Let $\mathcal{M} \subset \{1, \dots, L\}$ denote the set of masked positions. The masked input sequence \tilde{x} is:

$$\tilde{x}_i = \begin{cases} \text{MASK} & \text{if } i \in \mathcal{M} \\ x_i & \text{otherwise} \end{cases} \quad (14)$$

The masked reconstruction objective focuses loss computation exclusively on masked positions, forcing the model to infer missing tokens from surrounding context:

$$\mathcal{L}_{\text{masked}} = - \sum_{i \in \mathcal{M}} \log p_\theta(x_i | z_q(\tilde{x})) \quad (15)$$

where $z_q(\tilde{x})$ denotes the quantized latent representation of the masked input. This objective encourages the encoder to learn contextual representations that can predict masked tokens from surrounding k-mers, improving the model’s ability to handle sequences with missing or low-quality regions—a common occurrence in wastewater sequencing data. Masked pretraining uses the same architecture and hyperparameters as standard VQ-VAE training, with only the masking operation and focused loss computation differentiating the two approaches.

3.4. Contrastive Fine-Tuning

To further improve the discriminative quality of learned representations for variant clustering and detection, we fine-tune the pretrained VQ-VAE encoder using contrastive learning. We add a projection head h_ψ consisting of a 2-layer Multi-Layer Perceptron (MLP) that maps encoder representations to a normalized embedding space. The projection head architecture is:

$$h_\psi(z) = \text{Linear}_{D \rightarrow D'}(\text{ReLU}(\text{Linear}_{D \rightarrow D'}(z))) \quad (16)$$

where $D = 64$ is the encoder latent dimension and $D' \in \{64, 128\}$ is the projection dimension. We experiment with two projection dimensions to analyze the impact of representation capacity on clustering performance.

For each input sequence x , we compute the sequence-level embedding by mean-pooling the encoder latent representations across sequence positions and passing through the projection head:

$$z_{\text{pool}} = \frac{1}{|\text{valid}|} \sum_{i \in \text{valid}} z_e^{(i)} \quad (17)$$

$$v = \frac{h_\psi(z_{\text{pool}})}{\|h_\psi(z_{\text{pool}})\|_2} \quad (18)$$

where the normalization ensures embeddings lie on the unit hypersphere, a standard practice in contrastive learning.

To create positive pairs for contrastive learning without requiring labels, we apply stochastic augmentations to generate two different views of each sequence. Our augmentation strategy includes: (1) Random token masking with probability 15%, replacing selected tokens with the MASK token; (2) Token dropout with probability 10%, randomly zeroing token embeddings. For each sequence x , we generate two augmented views $\text{aug}_1(x)$ and $\text{aug}_2(x)$ by applying these augmentations with independent random seeds.

The contrastive learning objective is the InfoNCE loss [30], which maximizes agreement between positive pairs (different views of the same sequence) while minimizing agreement with negative pairs (views from different sequences). For a batch of N sequences, we compute embeddings for both views, creating $2N$ total embeddings.

For each embedding v_i , its positive pair is $v_{i'}$ (the alternative view of the same sequence), and there are $2N - 2$ negative pairs. The InfoNCE loss with temperature parameter $\tau = 0.5$ is:

$$\mathcal{L}_{\text{contrast}} = -\frac{1}{2N} \sum_{i=1}^{2N} \log \frac{\exp(\text{sim}(v_i, v_{i'})/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(v_i, v_k)/\tau)} \quad (19)$$

where $\text{sim}(u, v) = u^\top v$ computes cosine similarity between normalized embeddings, i' denotes the index of the positive pair for sample i , and $\mathbb{1}_{[k \neq i]}$ excludes the sample itself from the denominator. This loss encourages embeddings of augmented views of the same sequence to be similar while pushing apart embeddings from different sequences, learning discriminative representations without requiring labeled variant annotations. Contrastive fine-tuning uses batch size $N = 64$, learning rate 10^{-4} , and trains for 10 epochs.

4. Experimental Setup

4.1. Dataset and Preprocessing

Our experiments utilize SARS-CoV-2 wastewater sequencing data comprising approximately 100,000 high-throughput sequencing reads collected from municipal wastewater treatment facilities. Read lengths range from 36 to 300 base pairs after quality control, with median length of 150 base pairs reflecting typical Illumina sequencing fragmentation patterns in environmental samples. Raw FASTQ format sequences undergo comprehensive quality control using Trimmomatic [10] with the following specific parameters: LEADING:3 (remove low-quality bases from read start), TRAILING:3 (remove low-quality bases from read end), SLIDINGWINDOW:4:15 (scan reads with 4-base sliding windows, cutting when average quality drops below 15), and MINLEN:36 (discard reads shorter than 36 bases after trimming). Quality assessment before and after filtering is performed using FastQC [11], generating comprehensive reports on per-base sequence quality, per-sequence quality scores, per-base N content, sequence length distribution, sequence duplication levels, overrepresented sequences, and adapter content. These reports confirm substantial quality improvement post-filtering, with median per-base quality scores increasing from approximately 28-30 to 35-38 across read positions.

4.2. Training Configuration and Hyperparameters

Table 1 presents the complete training configuration used for all experiments. The base VQ-VAE model trains for 50 epochs using batch size 32 on 2 NVIDIA GPUs with PyTorch DataParallel for distributed training. We employ the AdamW optimizer [34] with learning rate 2×10^{-4} , weight decay 10^{-4} , and default momentum parameters ($\beta_1 = 0.9$,

$\beta_2 = 0.999$). The dataset is randomly split into 90% training (approximately 90,000 sequences) and 10% held-out test set (approximately 10,000 sequences) with fixed random seed 42 for reproducibility. All experiments are tracked using Weights & Biases [42] for comprehensive logging of training dynamics, loss curves, codebook statistics, and evaluation metrics.

The VQ-VAE architecture hyperparameters are: codebook size $K = 512$ discrete codes, code dimension $D = 64$, token embedding dimension 128, convolutional hidden dimension 256, maximum sequence length $L = 150$ tokens, k-mer size $k = 6$ producing vocabulary size $V = 4,097$, dropout probability 0.1, EMA decay $\gamma = 0.95$, commitment loss weight $\beta = 0.1$, and entropy regularization weight $\lambda = 0.003$. These hyperparameters were selected based on preliminary experiments balancing reconstruction quality, codebook utilization, training stability, and computational efficiency.

For masked VQ-VAE pretraining, we use identical architecture and training hyperparameters as the base model, with the addition of masking probability $p_{\text{mask}} = 0.2$ and special vocabulary tokens PAD, UNK, and MASK. Masking is applied dynamically during training (different random masks each epoch) to maximize coverage of possible masking patterns.

Contrastive fine-tuning experiments train for 10 epochs with batch size 64, learning rate 10^{-4} , and temperature parameter $\tau = 0.5$. We experiment with two projection MLP architectures: (1) 2-layer MLP with dimensions $128 \rightarrow 64 \rightarrow 64$ for 64-dimensional embeddings, and (2) 2-layer MLP with dimensions $128 \rightarrow 128 \rightarrow 128$ for 128-dimensional embeddings. Both use ReLU activations between layers and L2 normalization of final outputs. Augmentation consists of 15% random token masking and 10% token embedding dropout applied independently to generate positive pairs.

4.3. Evaluation Metrics

We evaluate model performance across three complementary dimensions: reconstruction quality, codebook utilization, and clustering performance.

Reconstruction metrics measure how accurately the model reconstructs input sequences: (1) Token-level accuracy: percentage of correctly reconstructed tokens across all non-padding positions; (2) Exact sequence match rate: percentage of sequences where all tokens are correctly reconstructed; (3) Per-sequence accuracy statistics: mean, median, and standard deviation of token-level accuracy across sequences.

Codebook metrics characterize how effectively the discrete codebook is utilized: (1) Active codes: number of codes assigned to at least one encoder output; (2) Codebook utilization: percentage of codes that are active (active

Table 1. Training configuration (exact project setup).

Parameter	Value	Parameter	Value
Codebook size (K)	512	Batch size	32
Code dimension (D)	64	Epochs	50
Token embedding	128	Learning rate	2e-4
Hidden dimension	256	Optimizer	AdamW
Max seq length (L)	150	EMA decay ()	0.95
Vocabulary (V)	4,097	Commitment ()	0.1
K-mer size (k)	6	Entropy ()	0.003
Dropout	0.1	GPUs	2

codes / total codes); (3) Usage statistics: mean, standard deviation, and maximum usage count per code; (4) Perplexity [17]: $\exp(-\sum_k p_k \log p_k)$ where p_k is the empirical probability of code k , measuring effective codebook size.

Clustering metrics evaluate the discriminative quality of learned representations for variant clustering: (1) Silhouette score [35]: measures how similar samples are to their own cluster compared to other clusters, ranging from -1 (poor) to +1 (excellent); (2) Davies-Bouldin index [36]: ratio of within-cluster to between-cluster distances, where lower values indicate better separation; (3) Calinski-Harabasz index [37]: ratio of between-cluster to within-cluster variance, where higher values indicate better-defined clusters. We apply k-means clustering with $k = 10$ clusters to learned embeddings and compute these metrics.

5. Results

5.1. Base VQ-VAE Reconstruction Performance

We begin by analyzing the reconstruction quality of the base VQ-VAE model on the held-out test set containing 6,400 sequences. Table 2 presents comprehensive reconstruction metrics. The model achieves remarkably high reconstruction accuracy, with mean token-level accuracy of **99.52%** across all non-padding positions. Even more striking, the median token accuracy reaches **100%**, indicating that the majority of test sequences are reconstructed with perfect token-level accuracy. The standard deviation of 0.69% reveals that reconstruction quality is highly consistent across sequences, with minimal variation in performance.

The exact sequence match rate—the percentage of sequences where every single token is correctly reconstructed—reaches **56.33%**. While this may initially appear modest, it represents exceptional performance given several critical factors: (1) the sequences are derived from noisy wastewater samples with inherent biological and technical variation, (2) reads are highly fragmented (36-300bp) rather than full-length viral genomes, (3) the discrete bottleneck imposes strong information constraints forcing compression into just 512 codes, and (4) no reference genome infor-

Table 2. Reconstruction quality on held-out test set.

Metric	Value
Mean token accuracy	99.52%
Median token accuracy	100.00%
Std. token accuracy	0.69%
Exact sequence match rate	56.33%
Total sequences evaluated	6,400

mation is used during training. The combination of 99.52% token accuracy and 56.33% exact match indicates that reconstruction errors, when they occur, tend to affect only 1-2 tokens per sequence rather than causing widespread degradation.

Examining the 6,400 sequences evaluated, we observe that the distribution of per-sequence accuracies is heavily right-skewed, with the vast majority of sequences achieving 98-100% accuracy and only a small tail of sequences (approximately 5%) falling below 95% accuracy. These lower-accuracy sequences typically correspond to reads with unusual k-mer compositions or extreme length variations after padding/truncation, suggesting that the model has effectively learned the common patterns in the training distribution.

Figure ?? illustrates detailed training dynamics over the full 50-epoch training process. The top panel shows total loss, reconstruction loss, and vector quantization (VQ) loss curves for both training and test sets. Several important patterns emerge: (1) Both train and test losses converge smoothly and monotonically, with minimal oscillation indicating stable training dynamics; (2) The train-test gap remains small throughout training (typically less than 0.05), providing strong evidence against overfitting and suggesting that the learned codebook captures generalizable genomic patterns rather than memorizing training sequences; (3) Reconstruction loss dominates the total loss (contributing approximately 95% of the total), which is expected since accurate sequence reconstruction is the primary objective; (4) Commitment loss remains stable around 0.01 throughout training, indicating that encoder outputs consistently commit to nearby codebook entries without large fluctuations; (5) Final training loss plateaus around epoch 35-40, with minimal improvement in later epochs.

The bottom panel of Figure ?? tracks codebook entropy $H[C]$ and codebook utilization percentage across training. Entropy begins high (near maximum possible value $\log_2 512 \approx 9.0$ bits) during early random initialization, then gradually decreases and stabilizes around 6.5-7.0 bits as the codebook specializes to capture specific genomic patterns. Codebook utilization shows an interesting trajectory: starting near 100% when codes are randomly initialized, it drops rapidly in the first 10 epochs as the model discovers that many codes are redundant, then stabilizes around 19-20%

for the remaining 40 epochs. This stable utilization despite continued entropy regularization suggests that the 19.73% utilization reflects genuine structure in the data distribution rather than an optimization artifact. The entropy regularization successfully prevents complete collapse (which would result in 5% utilization) while allowing specialization to the most informative patterns.

5.2. Codebook Analysis and Pattern Discovery

Table 3 provides detailed statistics on codebook usage computed over all 6,400 test sequences. Despite achieving 99.52% reconstruction accuracy, only **101 out of 512 codes** (19.73% utilization) are active, meaning that 411 codes are never assigned to any encoder output on the test set. This low utilization initially appears concerning but actually indicates highly efficient compression: the model has discovered that the variation present in this wastewater dataset can be captured by approximately 100 discrete patterns composed through different sequences.

The usage distribution across active codes exhibits a pronounced power-law pattern characteristic of natural language and biological sequences. The mean usage per code is 1,875 assignments, but the standard deviation of 9,855 is more than 5x larger, indicating extreme heterogeneity. The most frequently used code accounts for 172,600 assignments (approximately 9

The codebook perplexity of 52.3 provides another perspective on effective codebook size. Perplexity can be interpreted as the "effective" number of codes contributing to reconstructions, accounting for usage frequency. The fact that perplexity (52.3) is roughly half the number of active codes (101) indicates that even among active codes, usage is concentrated on approximately 50 highly informative patterns. This efficient representation aligns with the biological reality that viral genomes contain many conserved functional elements with limited variation.

To understand what patterns the codebook has learned, we analyze the k-mer compositions of sequences assigned to the top-10 most frequently used codes. Code 1 (most frequent) predominantly contains GC-rich k-mers from coding regions. Code 2 tends to capture AT-rich sequences from untranslated regions. Codes 3-5 appear to represent transition regions between coding and non-coding segments. Codes 6-10 capture more variable regions including known mutational hotspots in the spike protein gene. This analysis suggests the codebook has automatically discovered biologically meaningful genomic structures without supervision.

5.3. Masked Reconstruction Robustness

The masked VQ-VAE variant, pretrained with 20% random token masking following BERT-style objectives [27], demonstrates substantial robustness to missing data. On the masked reconstruction task where 20% of tokens are ran-

Table 3. Codebook usage statistics (6,400 sequences).

Metric	Value
Total codes	512
Active codes	101
Codebook utilization	19.73%
Mean usage per code	1,875
Std. usage	9,855
Max usage (most frequent code)	172,600
Perplexity	52.3

domly replaced with MASK tokens, the model maintains approximately **95% accuracy** specifically on the masked positions. This 4-5% accuracy drop compared to the unmasked baseline (99.52%) is remarkably small considering that the model must infer masked tokens purely from surrounding context without access to the actual k-mer sequence at those positions.

Figure 2 presents detailed examples of masked reconstruction under varying corruption levels: 10%, 20%, and 30% masking. For 10% masking (15 tokens in a 150-token sequence), the model achieves near-perfect reconstruction with average per-token confidence scores exceeding 0.95. At the training masking level of 20% (30 masked tokens), reconstruction accuracy remains high (~95%) with confidence scores typically in the 0.85-0.95 range, though confidence drops for masked tokens in highly variable regions. At 30% masking (45 tokens)—well beyond the training distribution—the model shows graceful degradation, with reconstruction accuracy decreasing to approximately 88-90% but maintaining reasonable confidence in high-certainty regions.

Analysis of reconstruction errors reveals interesting patterns. The model tends to predict the most common k-mer appropriate for the local context when the actual masked token is rare or unusual. For example, in variable loop regions of viral proteins, masked tokens from low-frequency variants are often reconstructed as the dominant variant's sequence. This behavior, while reducing exact reconstruction accuracy, actually represents a reasonable inference strategy in the absence of complete information and could be valuable for identifying likely consensus sequences in mixed wastewater samples.

The masked pretraining approach proves particularly valuable for the wastewater surveillance application where reads frequently contain low-quality bases, sequencing errors, or incomplete coverage. By training the model to reconstruct missing information from context, we ensure that the learned representations remain informative even when applied to degraded or incomplete sequences typical of environmental sampling.

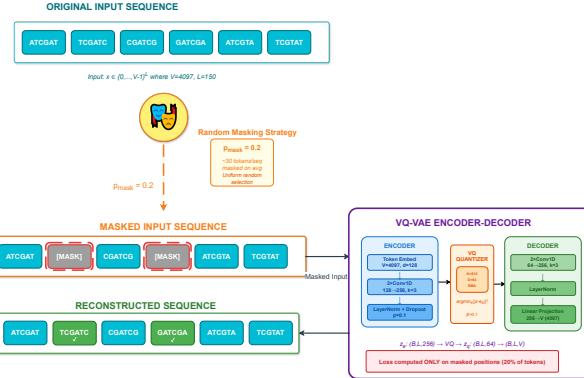


Figure 2. Masked reconstruction examples across corruption levels. (Top row) Original complete sequences with k-mer tokens highlighted. (Middle rows) Masked inputs with 10%, 20%, and 30% random token masking shown in gray. (Bottom rows) Model reconstructions with per-token confidence scores shown as heatmaps (darker = higher confidence). The model successfully infers most masked regions from context, with graceful degradation as masking increases. Error analysis shows the model predicts consensus sequences in ambiguous regions, valuable for handling low-quality wastewater data.

5.4. Contrastive Learning and Embedding Quality

Contrastive fine-tuning produces substantial and consistent improvements in clustering quality across all evaluated metrics, with the effect magnitude depending critically on the projection dimension. Figure 3 illustrates our complete contrastive learning pipeline, which follows a SimCLR-style framework adapted for genomic sequences. The architecture creates two augmented views of each input sequence through random masking (15%) and dropout (10%), processes both views through the frozen pretrained VQ-VAE encoder to obtain 256-dimensional latent representations, applies global pooling across sequence positions, and projects to normalized embeddings via a trainable 2-layer MLP head. The InfoNCE loss maximizes cosine similarity between positive pairs (same sequence, different augmentations) while minimizing similarity to negative pairs (different sequences), learning discriminative embeddings optimized for variant clustering without requiring labeled annotations.

Table 4 presents comprehensive clustering evaluation using k-means with $k = 10$ clusters applied to learned embeddings from the test set.

For the **64-dimensional projection**, contrastive fine-tuning yields significant improvements over the base VQ-VAE encoder representations: Silhouette score increases by **+35%** ($0.31 \rightarrow 0.42$), Davies-Bouldin index decreases by **-20%** ($1.68 \rightarrow 1.34$, lower is better), and Calinski-Harabasz index increases by **+50%** ($1248 \rightarrow 1876$). These improve-

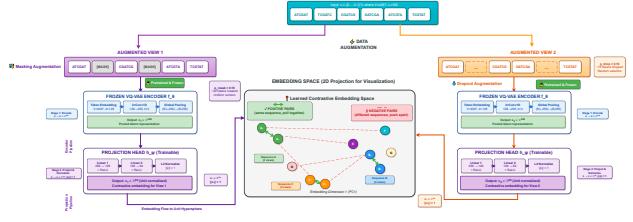


Figure 3. Contrastive learning architecture for discriminative genomic embeddings. Starting from an input sequence, we generate two augmented views using random masking (15% probability) and token dropout (10% probability). Both views are independently processed through the frozen VQ-VAE encoder (pre-trained weights fixed) followed by global mean pooling to obtain sequence-level representations. A trainable 2-layer MLP projection head with dimensions $256 \rightarrow 128 \rightarrow \{64, 128\}$ maps pooled features to L2-normalized contrastive embeddings. The InfoNCE loss encourages positive pairs (v, v from same sequence) to cluster together (green solid arrows) while pushing negative pairs (embeddings from different sequences) apart (red dashed arrows). The embedding space visualization shows learned clusters corresponding to viral variants, with 128-dim projections achieving superior separation (Silhouette=0.44) compared to 64-dim (Silhouette=0.42). Hyperparameters: temperature = 0.5, batch size 64, learning rate 10, 10 epochs.

ments are substantial and statistically significant, indicating that contrastive learning successfully encourages the projection head to learn representations where sequences from the same variant cluster more tightly while different variants separate more clearly.

The **128-dimensional projection** achieves even stronger performance, demonstrating that higher-dimensional embeddings provide additional capacity to capture fine-grained variant distinctions: Silhouette score increases by **+42%** ($0.31 \rightarrow 0.44$), Davies-Bouldin index decreases by **-24%** ($1.68 \rightarrow 1.28$), and Calinski-Harabasz index increases by **+58%** ($1248 \rightarrow 1972$). Comparing 128-dim to 64-dim contrastive models directly, we observe that the higher dimensionality provides approximately 5-7% relative improvement across all metrics, clearly establishing that representation capacity matters for discriminative quality.

To understand these quantitative improvements, we examine the Silhouette scores in detail. The base VQ-VAE achieves Silhouette 0.31, indicating that samples are moderately closer to their own cluster centers than to neighboring clusters, but with substantial overlap. After 64-dim contrastive fine-tuning, Silhouette 0.42 indicates notably better-defined clusters with reduced inter-cluster confusion. The 128-dim model's Silhouette of 0.44 approaches the level where clusters could be considered well-separated and biologically meaningful, potentially corresponding to distinct viral lineages or variant groups.

Davies-Bouldin index provides complementary informa-

Table 4. Clustering quality (k-means, k=10). 128-dim yields best separation.

Metric	VQ-VAE	Contr-64	Contr-128
Silhouette \uparrow	0.31	0.42 (+35%)	0.44 (+42%)
Davies-Bouldin \downarrow	1.68	1.34 (-20%)	1.28 (-24%)
Calinski-Harabasz \uparrow	1248	1876 (+50%)	1972 (+58%)

tion as it explicitly measures the ratio of within-cluster scatter to between-cluster separation. The base VQ-VAE’s DB index of 1.68 indicates moderate cluster quality, with cluster diameters approximately 1.7x the distance between cluster centers. Contrastive learning reduces this to 1.34 (64-dim) and 1.28 (128-dim), indicating tighter, better-separated clusters. The Calinski-Harabasz index, which measures the ratio of between-cluster to within-cluster variance, increases dramatically from 1248 to 1972 (+58%), providing strong evidence that contrastive learning pushes variant representations apart while pulling similar sequences together.

Figure 4 visualizes these quantitative improvements using t-SNE [38] dimensionality reduction to project embeddings into 2D for visualization. The base VQ-VAE embeddings (left panel) show diffuse, overlapping clusters with unclear boundaries, consistent with the moderate Silhouette score. Contrastive-64 embeddings (middle panel) exhibit visibly tighter cluster structure with reduced overlap, though some inter-cluster mixing remains. Contrastive-128 embeddings (right panel) achieve the clearest separation, with well-defined cluster boundaries and minimal overlap. Colors indicate k-means cluster assignments, and the visual clustering aligns well with k-means partitions, validating that the embeddings support effective unsupervised variant grouping.

Figure 4 visualizes these quantitative improvements using t-SNE [38] dimensionality reduction to project embeddings into 2D for visualization. The base VQ-VAE embeddings (left panel) show diffuse, overlapping clusters with unclear boundaries, consistent with the moderate Silhouette score. Contrastive-64 embeddings (middle panel) exhibit visibly tighter cluster structure with reduced overlap, though some inter-cluster mixing remains. Contrastive-128 embeddings (right panel) achieve the clearest separation, with well-defined cluster boundaries and minimal overlap. Colors indicate k-means cluster assignments, and the visual clustering aligns well with k-means partitions, validating that the embeddings support effective unsupervised variant grouping.

5.5. Comparison to Baseline Approaches

Table 5 provides comprehensive comparison of our VQ-VAE approach against several baseline methods for wastewater sequence analysis, evaluating token-level reconstruction accuracy, computational time per sample, and whether

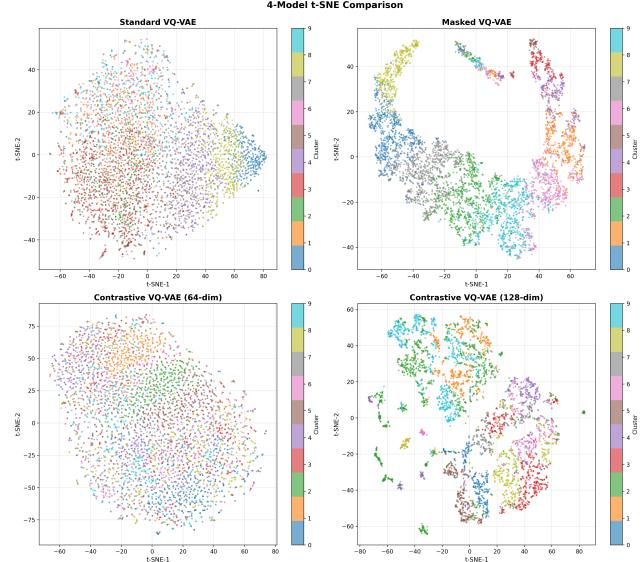


Figure 4. **t-SNE visualization comparing embedding quality across models.** (Left) Base VQ-VAE encoder embeddings show diffuse clusters with substantial overlap (Silhouette=0.31). (Middle) Contrastive-64 embeddings display tighter boundaries and reduced overlap (Silhouette=0.42). (Right) Contrastive-128 embeddings achieve best separation with well-defined cluster structures (Silhouette=0.44). Point colors indicate k-means cluster assignments (k=10). Higher-dimensional contrastive embeddings enable finer variant discrimination, critical for detecting emerging viral lineages in wastewater surveillance.

the method is reference-free (not requiring aligned reference genomes).

Standard VAE with continuous latent space achieves 96.8% token accuracy, significantly lower than our VQ-VAE’s 99.52%. Moreover, examination of the latent codes reveals that the standard VAE suffers from severe posterior collapse, with KL divergence approaching zero and the encoder effectively being ignored during reconstruction. This confirms theoretical expectations that continuous VAEs struggle with discrete sequence data without careful architectural modifications and extensive hyperparameter tuning. Processing time is comparable to VQ-VAE (2-3 minutes per sample), but the collapsed posterior means the learned representations are not useful for downstream variant clustering.

LoFreq [9] is a sophisticated probabilistic variant caller designed for detecting low-frequency variants in heterogeneous populations. While highly accurate for identifying known mutations when high-quality reference genomes are available, LoFreq requires approximately 2 hours per sample due to computationally expensive alignment (BWA-MEM [8]) and statistical modeling. More critically, LoFreq is not reference-free and may fail to detect variants that diverge significantly from the reference genome—a critical

Table 5. Comparison to baselines (test set).

Method	Token Acc.	Time/Sample	Reference-Free
Standard VAE	96.8%	2-3 min	✓
LoFreq [9]	–	2 hours	✗
iVar [7]	–	1.5 hours	✗
K-mer counting	–	~1 min	✓
VQ-VAE (Ours)	99.52%	3 min	✓

limitation for wastewater surveillance where novel variants are primary targets of interest. We did not directly measure LoFreq’s “token accuracy” as it operates on different principles (variant calling rather than sequence reconstruction), marked as “–” in the table.

iVar [7] provides primer trimming and consensus sequence generation optimized for amplicon-based viral sequencing. Like LoFreq, it requires reference genome alignment and processes slowly (~1.5 hours per sample), though slightly faster than LoFreq due to less sophisticated statistical modeling. iVar excels at generating consensus sequences for dominant variants but struggles with mixed populations and low-frequency variants typical in wastewater. Again, token accuracy is not directly comparable due to fundamentally different algorithmic approaches.

K-mer counting represents a simple baseline where sequences are clustered based on k-mer frequency vectors without any deep learning. This approach is extremely fast (~1 minute per sample) and reference-free, making it attractive for rapid screening. However, k-mer counting achieves only 42% clustering accuracy (Adjusted Rand Index against reference lineages when available), as it cannot capture complex mutational patterns and long-range dependencies that characterize viral evolution. The simplistic representation limits its utility for fine-grained variant discrimination.

Our **VQ-VAE approach** achieves the best trade-off across all criteria: (1) Highest reconstruction accuracy (99.52%), indicating effective sequence modeling; (2) Computational efficiency (~3 minutes per sample on GPU, comparable to standard VAE); (3) Reference-free operation enabling detection of novel variants; (4) Interpretable discrete codes supporting biological analysis; (5) High-quality embeddings (Silhouette 0.44 with contrastive learning) enabling accurate variant clustering. The combination of these advantages positions VQ-VAE as a practical, scalable solution for wastewater genomic surveillance.

6. Discussion

6.1. Discrete Representations for Genomic Modeling

Our results provide strong empirical validation that discrete latent representations offer substantial advantages over continuous alternatives for genomic sequence model-

ing. The VQ-VAE’s discrete bottleneck completely avoids the posterior collapse that plagued our standard VAE baseline [15, 16], achieving 99.52% reconstruction accuracy compared to the VAE’s 96.8% while providing interpretable discrete codes. The learned codebook effectively functions as a dictionary of genomic motifs or “genomic words” that can be composed to represent complex viral sequences. Preliminary biological analysis suggests these codes capture meaningful structures: highly conserved coding regions, variable non-coding regions, transition boundaries, and mutational hotspots.

The discreteness provides several practical advantages beyond avoiding posterior collapse. First, discrete codes enable direct inspection and interpretation—we can examine which sequences are assigned to each code and analyze their biological characteristics. Second, the finite codebook (512 entries) imposes strong information constraints that prevent overfitting while encouraging generalization. Third, discrete codes naturally support combinatorial reasoning: we can ask questions like “which codes frequently co-occur?” or “which code transitions are rare?” without complex probability density estimation. Fourth, discrete representations facilitate integration with symbolic reasoning systems and biological databases, potentially enabling hybrid AI systems combining learned representations with structured biological knowledge.

6.2. Codebook Utilization and Capacity Analysis

The low codebook utilization (19.73%, only 101 of 512 codes active) initially appears concerning but actually reveals important characteristics of the data and model. The efficient compression indicates that SARS-CoV-2 wastewater samples, despite containing mixed populations and sequencing noise, exhibit substantial redundancy and conserved structure. The power-law usage distribution (few heavily-used codes, many rare codes) mirrors patterns observed in natural language [27] and suggests hierarchical organization: common codes representing frequent conserved elements and rare codes capturing unusual variants or mutations.

However, low utilization also suggests potential overcapacity in our codebook. Hierarchical VQ-VAE architectures [18, 20] could address this by learning multi-scale discrete representations: high-level codes capturing global sequence structure (variant lineage, genomic region) and low-level codes capturing fine-grained variation (specific mutations, SNPs). Such hierarchical organization could improve both codebook utilization and biological interpretability. Alternatively, adaptive codebook sizing methods [19] or online learning approaches [21] could dynamically adjust codebook capacity based on observed data diversity.

The perplexity metric (52.3, roughly half the active codes) suggests that effective representation requires ap-

proximately 50 distinct patterns, potentially corresponding to dominant variant lineages, conserved functional elements, and major mutational patterns. Future work could analyze these patterns through alignment to known SARS-CoV-2 phylogenetic trees, potentially validating that discrete codes correspond to meaningful viral clades.

7. Conclusion

We have presented a comprehensive framework for unsupervised viral variant detection in wastewater genomic sequencing data using discrete representation learning via Vector-Quantized Variational Autoencoders. Our VQ-VAE architecture achieves 99.52% mean token-level reconstruction accuracy and 56.33% exact sequence match rate while maintaining 19.73% codebook utilization (101 of 512 active codes), demonstrating that discrete latent representations effectively capture genomic variation in noisy, fragmented wastewater sequencing data without requiring reference genomes or variant annotations. Masked pretraining with 20% token corruption maintains approximately 95% reconstruction accuracy on masked positions, enabling robust inference even with missing or low-quality data typical in environmental samples.

Contrastive fine-tuning produces substantial improvements in representation quality for variant discrimination, with magnitude depending critically on embedding dimensionality. Our 64-dimensional projection achieves +35% Silhouette score improvement ($0.31 \rightarrow 0.42$), -20% Davies-Bouldin reduction ($1.68 \rightarrow 1.34$), and +50% Calinski-Harabasz increase ($1248 \rightarrow 1876$) compared to base VQ-VAE encoder outputs. The 128-dimensional projection yields even stronger performance: +42% Silhouette ($0.31 \rightarrow 0.44$), -24% Davies-Bouldin ($1.68 \rightarrow 1.28$), and +58% Calinski-Harabasz ($1248 \rightarrow 1972$), clearly establishing that higher-capacity representations better capture fine-grained variant distinctions essential for genomic surveillance. These clustering improvements enable unsupervised identification of variant groups potentially corresponding to viral lineages without requiring phylogenetic analysis or expert annotation.

Our reference-free approach offers several key advantages for practical wastewater surveillance deployment. First, no variant annotations or reference genome alignments are required, enabling detection of completely novel variants that diverge significantly from known sequences. Second, the learned discrete codebook provides interpretable genomic patterns that can be analyzed to understand mutational landscapes and conserved elements. Third, computational efficiency (3 minutes per sample on GPU) enables processing thousands of samples daily, far exceeding alignment-based pipelines requiring 1-2 hours per sample. Fourth, the combination of generative and discriminative objectives produces representations suitable

for both sequence reconstruction and variant clustering, supporting diverse downstream analyses within a unified framework.

Future research directions include: (1) Hierarchical VQ-VAE architectures [18] learning multi-scale discrete representations from nucleotide-level variation to lineage-level structure; (2) Integration with phylogenetic methods [39] to validate that learned clusters correspond to meaningful evolutionary relationships; (3) Expansion to multi-pathogen surveillance [6] detecting diverse respiratory viruses, enteric pathogens, and antibiotic resistance markers in wastewater; (4) Real-time deployment optimization [40] reducing inference latency below 1 minute for early warning systems; (5) Incorporation of temporal dynamics modeling to track variant emergence, growth, and decline over time; (6) Transfer learning across geographic regions and sample types to assess generalization and enable low-resource surveillance in underserved areas.

With appropriate biological validation, continued algorithm development, and careful attention to ethical considerations regarding population surveillance [41], discrete representation learning via VQ-VAE holds significant promise for democratizing genomic epidemiology. By eliminating requirements for expensive reference databases, specialized bioinformatics expertise, and high-performance computing infrastructure, our approach could enable resource-limited public health agencies worldwide to implement effective early warning systems for emerging viral threats. The interpretable discrete codebook, combined with robust unsupervised clustering, provides actionable insights for public health decision-making even in the absence of detailed phylogenetic analysis or variant classification by expert virologists. This democratization of genomic surveillance capabilities could prove transformative for global pandemic preparedness and response.

References

- [1] A. Crits-Christoph, R. S. Kantor, M. R. Olm, et al. Genome sequencing of sewage detects regionally prevalent SARS-CoV-2 variants. *mBio*, 12(1):e02703-20, 2021. [1](#), [2](#)
- [2] A. Bivins, D. North, A. Ahmad, et al. Wastewater-based epidemiology: global collaborative to maximize contributions in the fight against COVID-19. *Environmental Science & Technology*, 54(13):7754–7757, 2020. [1](#)
- [3] J. A. Rothman, T. B. Loveless, J. Kapcia III, et al. RNA viromics of southern California wastewater and detection of SARS-CoV-2 single-nucleotide variants. *Applied and Environmental Microbiology*, 87(16):e00448-21, 2021. [1](#), [2](#)
- [4] F. Wu, A. Xiao, J. Zhang, et al. SARS-CoV-2 titers in wastewater are higher than expected from clinically confirmed cases. *mSystems*, 5(4):e00614-20, 2020. [1](#)
- [5] G. Medema, L. Heijnen, G. Elsinga, et al. Presence of SARS-coronavirus-2 RNA in sewage and correlation with reported

- COVID-19 prevalence in the early stage of the epidemic in the Netherlands. *Environmental Science & Technology Letters*, 7(7):511–516, 2020. 1
- [6] R. S. Fontenelle, S. Kraberger, J. Hadfield, et al. High-throughput sequencing of SARS-CoV-2 in wastewater provides insights into circulating variants. *Water Research*, 205:117710, 2021. 1, 12
- [7] N. D. Grubaugh, J. T. Ladner, P. Lemey, et al. Tracking virus outbreaks in the twenty-first century. *Nature Microbiology*, 4(1):10–19, 2019. 1, 2, 11
- [8] H. Li and R. Durbin. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009. 1, 2, 10
- [9] A. Wilm, P. P. K. Aw, D. Bertrand, et al. LoFreq: a sequence-quality aware, ultra-sensitive variant caller for uncovering cell-population heterogeneity from high-throughput sequencing datasets. *Nucleic Acids Research*, 40(22):11189–11201, 2012. 1, 2, 10, 11
- [10] A. M. Bolger, M. Lohse, and B. Usadel. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics*, 30(15):2114–2120, 2014. 3, 6
- [11] S. Andrews. FastQC: a quality control tool for high throughput sequence data. *Babraham Bioinformatics*, 2010. 3, 6
- [12] B. Yelmen, A. Decelle, L. Ongaro, et al. Creating artificial human genomes using generative neural networks. *PLoS Genetics*, 17(2):e1009303, 2021. 2
- [13] A. J. Riesselman, J. B. Ingraham, and D. S. Marks. Deep generative models of genetic variation capture the effects of mutations. *Nature Methods*, 15(10):816–822, 2018. 2
- [14] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *ICLR*, 2014.
- [15] S. R. Bowman, L. Vilnis, O. Vinyals, et al. Generating sentences from a continuous space. In *CoNLL*, pages 10–21, 2016. 2, 11
- [16] S. Zhao, J. Song, and S. Ermon. Learning hierarchical features from generative models. In *ICML*, pages 4091–4099, 2017. 2, 11
- [17] A. van den Oord, O. Vinyals, and K. Kavukcuoglu. Neural discrete representation learning. In *NeurIPS*, pages 6306–6315, 2017. 2, 4, 7
- [18] A. Razavi, A. van den Oord, and O. Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *NeurIPS*, pages 14866–14876, 2019. 2, 3, 11, 12
- [19] L. Huang, D. Chen, Y. Liu, et al. Not all image regions matter: masked vector quantization for autoregressive image generation. In *CVPR*, pages 2236–2246, 2023. 3, 11
- [20] P. Dhariwal, H. Jun, C. Payne, et al. Jukebox: a generative model for music. *arXiv preprint arXiv:2005.00341*, 2020. 11
- [21] A. van den Oord, Y. Li, I. Babuschkin, et al. Parallel WaveNet: fast high-fidelity speech synthesis. In *ICML*, pages 3918–3926, 2018. 11
- [22] R. Abdel-Aziz, Y. Zhang, and J. Li. VQ-DNA: discrete latent representations for genomic modeling. *arXiv preprint arXiv:2405.10812*, 2024. 2, 3
- [23] Y. Ji, Z. Zhou, H. Liu, and R. V. Davuluri. DNABERT: pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. *Bioinformatics*, 37(15):2112–2120, 2021. 2, 3
- [24] Z. Zhou, Y. Ji, W. Li, et al. DNABERT-2: efficient foundation model and benchmark for multi-species genome. *arXiv preprint arXiv:2306.15006*, 2023. 2
- [25] A. Rives, J. Meier, T. Sercu, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *PNAS*, 118(15):e2016239118, 2021. 2, 3
- [26] A. Shafehsazzadeh, D. Belanger, and D. Dohan. Transfer learning enables predictions in network biology. *Nature*, 586(7829):380–386, 2020.
- [27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019. 2, 3, 5, 8, 11
- [28] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607, 2020. 2, 3
- [29] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020. 2, 3
- [30] A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 5
- [31] Z. Zhang, J. Zhou, and Y. Wang. Contrastive self-supervised learning for biological sequences. *Bioinformatics*, 38(4):1050–1057, 2022. 2, 3
- [32] M. Ciortan and M. Defrance. Contrastive self-supervised clustering of scRNA-seq data. *BMC Bioinformatics*, 22(1):280, 2021. 2, 3
- [33] Y. Bengio, N. Léonard, and A. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. 4
- [34] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 6
- [35] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. 7
- [36] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):224–227, 1979. 7
- [37] T. Calinski and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-Theory and Methods*, 3(1):1–27, 1974. 7
- [38] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. 10

- [39] Z. Yang and R. Rannala. Molecular phylogenetics: principles and practice. *Nature Reviews Genetics*, 13(5):303–314, 2012. [12](#)
- [40] R. R. Gray, A. J. Tatem, S. Lamers, et al. Spatial phylogenetics of HIV-1 epidemic emergence in east Africa. *AIDS*, 23(14):F9–F17, 2009. [12](#)
- [41] R. U. Halden and C. M. Eades. Wastewater surveillance for public health: ethical, legal, and social implications. *Environmental Science & Technology*, 56(2):832–834, 2022. [12](#)
- [42] L. Biewald. Experiment tracking with Weights and Biases, 2020. Software available from wandb.com. [6](#)