SHIBAURA INSTITUTE OF TECHNOLOGY

The Improved Speech Spectral Envelope Compression Based on VQ-VAE with Adversarial Technique

by
Tanasan Srikotr

A dissertation submitted in partial fulfillment
for the degree of Doctor of Engineering

in the
Division of Functional Control Systems
Graduate School of Engineering and Science

September 2022

“It always seems impossible until it's done."

Nelson Mandela

Acknowledgments

Though my name appears on the front cover of this dissertation, all the compliments and my gratitude must go to Professor Dr. Kazunori Mano. My best supervisor throughout my years at Shibaura Institute of Technology. I have learned a lot, yet I still feel there is so much more I could learn from him. He is a man of vision and intelligence, and I am fortunate to have been his student for the four years of my time at Shibaura Institute of Technology. Whenever I passed through a difficult time, he provided the light in the darkest night. Furthermore, I express my further thanks to all the Professors of my thesis defense committee, Dr. Tetsunori Kobayashi, Dr. Masanobu Takahashi, Dr. Nicodimus Retdian, and Dr. Eri Ioka, for their valuable discussions and comments.

I am grateful to all my family and friends in my homeland. Being away from home was never easy, and they have greatly supported that. They have made sure that I have nothing to worry about at home. They have ensured that I can focus 100% on my journey and my dream. Moreover, for that, I can never be thankful enough.

My further thanks must go to all my friends of all nationalities in Japan. They have been supportive in all situations. They are my laughter, my comfort corner, and my smiles.

My special gratitude goes down to the Hybrid Twinning Program of Shibaura Institute of Technology and all the university staff that always help me a lot on the journey trip to international conferences. They all have provided enormous support for me to develop cutting-edge research.

Finally, I would like to thank everyone at Professor Mano's laboratory of SIT. It was a joy and unforgettable memory working with everyone in the laboratory.

Tanasan Srikotr

SHIBAURA INSTITUTE OF TECHNOLOGY

Abstract

Graduate School of Engineering and Science
Division of Functional Control Systems

Doctor of Engineering

by Tanasan Srikotr

The speech vocoder or speech coding is designed to reduce the speech information in the sender before transmitting it via transmission media to the receiver and to reproduce the speech information. This kind of methodology saves the transmission system bandwidth and increases the number of users in the transmission system. However, we must carefully design the speech vocoder; otherwise, the quality of the reconstructed speech waveform deteriorates.

The famous speech coding is the Linear Predictive Coding (LPC-10) and Code-Excited Linear Prediction (CELP) based on speech analysis and speech synthesis system. Spectral envelopes are the critical speech parameter in speech processing. However, many methods based on Cepstrum and LPC cannot always synthesize natural-sounding speech. This dissertation extracts the high-quality spectral envelope from the WORLD vocoder to examine the speech quantization performance based on deep learning — the full spectral envelopes estimated from the WORLD vocoder can synthesize the high-fidelity speech waveform. However, the spectral envelopes are hard to quantify to obtain the quantized spectral envelopes acceptable to synthesize the natural, high-quality speech waveform.

The proper conventional compression technique required to quantize the spectral envelope parameters is Vector Quantization (VQ). Lately, deep learning technologies have shown an advantage compared to conventional VQ. The Vector Quantized Variational AutoEncoder

(VQ-VAE) is an end-to-end compression technique based on the deep learning method. The VQ-VAE is the quantization version of the Variational AutoEncoder (VAE). The difference between a VAE and a VQ-VAE is that the VAE learns continuous z-latent representations, whereas the VQ-VAE learns discrete z-latent representations. The compression based on deep learning widely introduces the VQ-VAE because the VQ-VAE provides better performance than conventional VQ methods such as LBG or K-means.

This dissertation's whole study focuses on the advantage of deep learning in reducing the reconstruction errors of speech spectral envelope quantization compared to the conventional VQ and the VQ-VAE.

The first part of the study in this dissertation examined the effect of deep learning architecture on VQ based on deep learning. The conventional VQ and the VQ based on deep learning were compared for the spectral envelope quantization performance. The spectral envelope parameters were extracted from a high-quality vocoder named WORLD at 48 kHz sampling frequency in the experiments. The quantization performance in four target bitrate operations varied from low to high bitrates was evaluated. We proposed the Multi-layers Perceptron Vector Quantized Variation AutoEncoder (MLP-VQ-VAE). It reduced the memory sizes of z-latent representations and embedding space (codebook) by around 1.6 times compared to the conventional VQ and 21.4 times for the VQ-VAE. It also decreased the average Log Spectral Distortion (LSD) by around 1.1 points in dB lower than the conventional VQ and around 2.5 points in dB than the VQ-VAE.

The second study was about the techniques of VQ in VQ-VAE and investigated the possibility of improving the reconstruction performance. We proposed the Sub-band Vector Quantized Variational AutoEncoder (Sub-band VQ-VAE) and the Predictive Vector Quantized Variational AutoEncoder (Predictive VQ-VAE). The spectral envelope quantization performance of the WORLD vocoder at 48 kHz sampling frequency was compared. The experimental results

for the four target bitrates showed that the Sub-band VQ-VAE reduced the average LSD by around 1.3 points in dB compared to the conventional VQ-VAE. The Predictive VQ-VAE results indicated that it had a lower distortion in terms of LSD than the VQ-VAE, about 2.58 points in dB for the four target bitrates.

The last study in the dissertation was the advanced deep learning training technique in VQ-VAE. The collaborative design of the VQ-VAE and the Generative Adversarial Network (GAN) worked together in the spectral envelope quantization of the WORLD vocoder, operated at 16 kHz sampling frequency. We proposed the three different methods in deep learning trainings with GAN architectures: the VAEGAN implemented in VQ-VAE, the VQ-VAE-EMGAN, and the VQ-VAE-EMDEC. They were compared with the VQ-VAE for the quantization performance in four target bitrate operations. The proposed training methods with GAN showed the effectiveness. The VQ-VAE-EMDEC reduced the average LSD by around 0.98 points in dB, the average L2 z-latent error by around 0.11 and in terms of reconstructed speech waveform, it also improved the Perceptual Evaluation Speech Quality (PESQ) by around 0.32 compared to the VQ-VAE.

Contents

Acknowledgments ii
Abstract iii
List of Figures x
List of Tables xiii
Chapter 1 Introduction 1
1.1 The speech compression 1
1.2 Objectives 4
1.3 Structure of this dissertation 4
Chapter 2 Related Studies 8
2.1 The speech vocoder 8
2.2 The Vector Quantization 11
2.2.1 Conventional Vector Quantization 11
2.2.2 Sub-band Vector Quantization 11
2.2.3 Predictive Vector Quantization 12
2.3 Deep learning 13
2.3.1 The AutoEncoder 13
2.3.2 The Variational AutoEncoder 15
2.3.3 The Vector Quantized Variational AutoEncoder 18
2.3.4 The Generative Adversarial Networks 22
2.4 Perceptual Evaluation of Speech Quality 23
2.4.1 Subjective Quality Assessments 23
2.4.2 Objective Quality Assessments 23
2.4.2.1 Mean Square Error 23
2.4.2.2 L2 Error 25
2.4.2.3 Log Spectral Distortion 25
2.4.2.4 Perceptual Evaluation of Speech Quality 25
Chapter 3 Overview of speech spectral envelope quantization based on deep learning 26
3.1 Overview 26
3.2 Scope of the Spectral Envelope Quantization Scope 27
Chapter 4 The vector quantization based on deep learning for speech spectral envelope quantization 29
4.1 Overview 29
4.2 Methodology 30
4.2.1 Conventional Vector Quantization 30
4.2.2 Vector Quantized Variational AutoEncoder (VQ-VAE) 31

4.2.3 Multilayer Perceptron Vector Quantized Variational AutoEncoder (MLP-VQ-VAE) 32
4.3 Experiments and Results 33
4.3.1 WORLD vocoder spectral envelope quantization 33
4.3.2 Raw speech waveform data base 34
4.3.3 The comparison of MLP-VQ-VAE and conventional vector quantization for spectral envelope quantization 34
4.3.4 The comparison of MLP-VQ-VAE and VQ-VAE for spectral envelope quantization 44
4.4 Discussion 53
4.5 Conclusion 53
Chapter 5 The effect of vector quantization techniques in vector quantization based on deep learning for speech spectral envelope quantization 55
5.1 Overview 55
5.2 Methodology 56
5.2.1 Vector Quantized Variational AutoEncoder (VQ-VAE) 56
5.2.2 The sub-band Vector Quantized Variational AutoEncoder (Sub-band VQ-VAE) 58
5.2.3 Predictive Vector Quantization Variational AutoEncoder (Predictive VQ-VAE) 59
5.3 Experiments and Results 60
5.3.1 WORLD vocoder spectral envelope quantization 60
5.3.2 Raw speech waveform data base 61
5.3.3 The sub-band Vector Quantized Variational AutoEncoder for Spectral Envelope Quantization 61
5.3.4 The Predictive Vector Quantized Variational AutoEncoder for Spectral Envelope Quantization 70
5.4 Discussion 83
5.4.1 The Sub-band Vector Quantized Variational AutoEncoder for Spectral Envelope Quantization 83
5.4.2 The Predictive Vector Quantized Variational AutoEncoder for Spectral Envelope Quantization 83
5.5 Conclusion 84

Chapter 6 The effect of deep learning network architecture and training techniques for speech spectral envelope quantization 85
6.1 Overview 85
6.2 Methodology 86
6.2.1 Vector Quantized Variational AutoEncoder (VQ-VAE) 86
6.2.2 Generative Adversarial Networks (GAN) 86
6.2.3 Deep learning parameter optimization 87
6.2.3.1 The gradient descent with the momentum algorithm 87
6.2.3.2 Root Mean Square Propagation 88
6.2.3.3 Adam optimization 88
6.2.4 Variational AutoEncoder Generative Adversarial Networks (VAEGAN) 89
6.2.5 The VAEGAN implemented in the Vector Quantized Variational AutoEncoder (VAE-GAN implemented in VQ-VAE) 89
6.2.6 The Vector Quantized Variational AutoEncoder with Embedding space Learned by Generative Adversarial Networks 95
6.2.7 The Vector Quantized Variational AutoEncoder with Embedding space and Decoder Network Learned by Generative Adversarial Networks. 98
6.3 Experiments and Results 101
6.3.1 The raw speech waveform database 101
6.3.2 The spectral envelope parameter quantization 101
6.3.3 The VQ-VAE, VAEGAN implemented in VQ-VAE, VQ-VAE-EMGAN, and the VQ-VAE-EMDEC implementation 111
6.3.4 The results of the performance comparison 113
6.3.5 The effects of the model parameter initialization 128
6.4 Discussion 131
6.5 Conclusion 133
Chapter 7 Conclusion and Future Work 134
7.1 Conclusion 134
7.2 Future Work 137
Appendix A 138
List of Publications and Awards 138
A.1 International journal paper 138
A.2 International conference papers 138

A.3 International conference paper (non-reviewed) 138
A.4 Awards and Scholarships 138
Bibliography 140

List of Figures

2.1	(a) Linear Prediction Coding (LPC-10)	9
2.1	(b) Code-Excited Linear Prediction (CELP)	9
2.2	(a) The WORLD vocoder	10
2.2	(b) The spectral envelope estimation comparison, DFT, Mel-cepstral spectral envelope, LPC-10 spectral envelope, and the WORLD vocoder spectral envelope	10
2.3	The sub-band Vector Quantization	12
2.4	The Predictive Vector Quantization	13
2.5	AutoEncoder	14
2.6	VAE z-latent estimation (Normal distribution)	15
2.7	VAE with the multivariate Gaussian assumption.	17
2.8	Vector Quantized Variational AutoEncoder	18
2.9	Embedding space e	20
2.10	$z_e(x)$, reshaped $z_e(x)$, and Embedding space	20
2.11	Embedding space, EMA, and EMA cluster size	21
2.12	The Generative Adversarial Networks	23
3.1	The WORLD vocoder Spectral Envelope Quantization	27
4.1	Overview of the well-known conventional vector quantization compared to the vector quantization based on deep learning for speech spectral envelope quantization	30
4.2	The block diagram of MLP-VQ-VAE	32
4.3	The Conventional VQ quantization diagram	36
4.4	The Conventional VQ Training process	36
4.5	The proposed MLP-VQ-VAE diagram	39
4.6	The proposed MLP-VQ-VAE training process	39
4.7	The conventional VQ average LSD evaluation	41
4.8	The proposed MLP-VQ-VAE average LSD evaluation	41
4.9	The comparison of LSD (dB) in four targets of bits/SP vector, the conventional vector quantization and the proposed MLP-VQ-VAE	42
4.10	The example of quantized spectral envelope parameter frames; (a), (c), (e), (g) are 45, 255, 369, 1845 bits/SP vector of conventional vector quantization, respectively. (b), (d), (f), (h) are 45, 255, 369, 1845 bits/SP vector of MLP-VQ-VAE, respectively	43

4.11	The VQ-VAE diagram	45
4.12	The VQ-VAE training process	46
4.13	The proposed MLP-VQ-VAE diagram	48
4.14	The proposed MLP-VQ-VAE training process	48
4.15	The VQ-VAE average LSD evaluation	50
4.16	The proposed MLP-VQ-VAE average LSD evaluation	50
4.17	The comparison of LSD (dB) in four targets of bits/SP vector, the VQ-VAE and the proposed MLP-VQ-VAE.	51
4.18	The example of quantized spectral envelope parameter frames. (a), (c), (e), (g) are 72, 288, 576, 2313 bits/SP vector of VQ-VAE, and (b), (d), (f), (h) are 72, 288, 576, 2313 bits/SP vector of MLP-VQ-VAE, respectively.	52
5.1	Overview of the proposed Sub-band VQ-VAE and the Predictive VQ-VAE compared to the VQ-VAE for WORLD vocoder spectral envelope quantization	56
5.2	The VQ-VAE model	57
5.3	The sub-band Vector Quantized Variational AutoEncoder	59
5.4	The Predictive Vector Quantized Variational AutoEncoder	60
5.5	The VQ-VAE quantization process	63
5.6	The Sub-band VQ-VAE quantization process	64
5.7	The VQ-VAE training process	66
5.8	The Sub-band VQ-VAE training process	66
5.9	The VQ-VAE average LSD evaluation	69
5.10	The proposed Sub-band VQ-VAE average LSD evaluation	69
5.11	The comparison of LSD (in dB) in 4 target bitrates.	70
5.12	The VQ-VAE quantization process for the comparison with the Predictive VQ-VAE	77
5.13	The VQ-VAE training process for the comparison with the Predictive VQ-VAE	77
5.14	The Predictive VQ-VAE quantization process	78
5.15	The Predictive VQ-VAE training process	78
5.16	The VQ-VAE average LSD evaluation	80
5.17	The proposed Predictive VQ-VAE average LSD evaluation	80
5.18	The comparison of average LSD results in four target bitrates	81
5.19	The comparison of quantized SP vectors in four target bitrates; (a) is the VQ-VAE and (b) is the Predictive VQ-VAE.	82
6.1	The procedure of VAEGAN implemented in VQ-VAE algorithm	90
6.2	The procedure of the VQ-VAE-EMGAN algorithm.	97
6.3	The procedure of VQ-VAE-EMDEC algorithm	100

6.4 The block diagram of the WORLD spectral envelope parameter quantization 103
6.5 The VQ-VAE diagram. 105
6.6 The VQ-VAE training process 106
6.7 The quantization diagram of the VAEGAN implemented VQVAE, the VQ-VAE-EMGAN, and the VQ-VAE-EMDEC. 107
6.8 The VAEGAN implemented in VQ-VAE training process 108
6.9 The VQ-VAE-EMGAN training process 109
6.10 The VQ-VAE-EMDEC training process 110
6.11 The VQ-VAE average of LSD, z-latent L2 error, and the PESQ score evaluation 114
6.12 The proposed VAEGAN implemented in VQ-VAE average of LSD, z-latent L2 error, and the PESQ score evaluation 114
6.13 The proposed VQ-VAE-EMGAN average of LSD, z-latent L2 error, and the PESQ score evaluation 115
6.14 The proposed VQ-VAE-EMDEC average of LSD, z-latent L2 error, and the PESQ score evaluation 115
6.15 The varied bit/SP vector comparison results of z-latent L2 error 118
6.16 The varied bit/SP vector comparison results of LSD 118
6.17 The varied bit/SP vector comparison results of PESQ 119
6.18 The loss term comparison results 119
6.19 The sampled z-latent comparison between unquantized and quantized methods at the operation bitrate of 1024 Bits/SP. 120
6.20 The sampled WORLD vocoder phoneme spectral envelope frames comparison 1024 bits/SP 121
6.21 The sampled WORLD vocoder phoneme spectral envelope frames comparison 512 bits/SP 122
6.22 The sampled WORLD vocoder phoneme spectral envelope frames comparison 256 bits/SP 122
6.23 The sampled WORLD vocoder phoneme spectral envelope frames comparison 128 bits/SP 123
6.24 The sampled WORLD vocoder spectral envelope (spectrogram) at 128 bits/SP 124
6.25 The sampled WORLD vocoder spectral envelope (spectrogram) at 256 bits/SP 125
6.26 The sampled WORLD vocoder spectral envelope (spectrogram) at 512 bits/SP 126
6.27 The sampled WORLD vocoder spectral envelope (spectrogram) at 1024 bits/SP 127

List of Tables

2.1	Mean Opinion Score (MOS)	24
4.1	The implementation of four conventional vector quantization models	34
4.2	The implementation of four MLP-VQ-VAE models for comparison with conventional vector quantization	38
4.3	The implementation of four VQ-VAE models	45
4.4	The implementation of four MLP-VQ-VAE models for compression with VQ-VAE	47
5.1	The VQ-VAE Architecture used for Sub-band VQ-VAE	63
5.2	The Implemented four model comparison	67
5.3	The VQ-VAE architecture for the comparison of the Predictive VQ-VAE	72
5.4	The Predictive VQ-VAE architecture	73
5.5	The four VQ-VAE implementation models for the comparison with Predictive VQ-VAE	76
5.6	The four Predictive VQ-VAE implementation models	76
6.1	The encoder network and decoder network architectures of four implemented techniques	104
6.2	The discriminator network architecture of the VAEGAN implemented in VQVAE, the VQ-VAE-EMGAN, and the VQ-VAE-EMDEC.	111
6.3	The four implementation models for the VQ-VAE, the VAEGAN implemented in VQ-VAE, the VQ-VAE-EMGAN, and the VQ-VAE-EMDEC.	112
6.4	The comparison results	117
6.5	The VQ-VAE-EMGAN model 1 initialization method comparison results	130
6.6	The VQ-VAE-EMGAN model 1 repeat model parameter initialization comparison results	130

Chapter 1

Introduction

1.1 The speech compression

In digital signal processing [1, 2, 3], speech compression [4, 5, 6, 7] is the technique to reduce the speech information at the encoder before transmitting, and the speech decompression reproduces the speech information from the reduced data at the decoder. This kind of application can save the digital transmission bandwidth [8, 9, 10, 11], the power consumption of the system, and increase the number of users in the transmission system for the world society [12, 13, 14]. However, we must carefully design those compression techniques because some speech parameters are susceptible to quantization distortion [15, 16, 17]. If a quantizer is not good enough, the quantized speech parameters include the distortion and impact to reconstruct the output speech waveform. The human speech waveforms are transformed from the analog signal to the digital signal by Analog to Digital Converter (Pulse Code Modulation techniques) [18, 19, 20] such as A-law and μ-law algorithms (G.711) [21, 22, 23] as speech encoding. The transformed digital speech signal is called a raw speech signal and is applied to the speech coding or vocoder [24, 25] to reduce the speech information before sending the transmission system. The received of the reduced data applied the speech decoding technique to represent the raw speech again in digital speech signal representation. Finally, the Digital to Analog Converter [26, 27] is applied to the digital signal to reproduce the speech signal at the end.

The famous speech vocoder and encoder are the Linear Predictive Coding (LPC-10) [28, 29, 30] and Code-Excited Linear Prediction (CELP) [31, 32, 33] based on speech analysis and speech synthesis systems. First, the raw speech waveform is analyzed to extract speech parameters such as the fundamental frequency (F0), aperiodicity (AP), and the spectral envelope (SP) during the speech analysis process. Then, those parameters are applied to reproduce the raw speech parameters during the synthesis process. At the intermediate, the compression techniques compress the extracted speech parameters before transmitting, and the decompression techniques reproduce the speech parameters. The most valuable parameter and most complex parameter for speech synthesis is the SP parameter [34, 35] compared to the F0 and AP because SP is very sensitive to distortions; if the compressor or

quantizer is not good enough, the reconstructed SP parameter's quality is low and leads to produce the low-quality speech waveform.

Speech coding techniques are widely used in speech communication systems [36, 37], especially in wireless cellular phone systems [38, 39]. The codec algorithms give complete performances through many studies in coding technologies. The speech vocoder is one of the fundamental compression techniques to reduce human speech data at a low bitrate. The spectral envelope parameter is the critical speech parameter for the vocoder's quality and intelligibility [34, 35]. Therefore, the proper conventional compression technique is required to quantize the spectral envelope parameter. Vector Quantization (VQ) [40, 41] is the conventional compression method for spectral envelope parameter quantization, such as the LPC-10 vocoder [28, 29, 30].

Lately, deep learning has shown an advantage compared to conventional VQ. In speech processing, compression technologies have been applied to reduce the amount of speech data in the limited communication bandwidth for information transmissions. One of the classic compression methods used in speech processing is Vector Quantization (VQ). The VQ method uses Linde–Buzo–Gray (LBG) or K-means algorithms [40, 41]. Some research studies have been conducted about deep learning for VQ. Deep learning Vector Quantization (DLVQ) [42] is introduced as a method for acoustic information retrieval. This model learns the code-constrained embedding space and performs better than the conventional VQ method in classification problems. In another research shown in [43], the mixture of deep learning and the VQ method has shown improvement. The model applied VQ as an encoder network and constructed the decoder network with deep learning networks. The deep learning decoder network has given the state-of-the-art method compared to the conventional inverse VQ as the decoder. As mentioned in these examples, the deep learning approach improved the performance of the VQ method.

The Vector Quantized Variational AutoEncoder (VQ-VAE) [44, 45] has been proposed as an end-to-end VQ method based on deep learning. The VQ-VAE architecture constructs of AutoEncoder (AE) [46, 47, 48, 49, 50] cooperated with the VQ method [40, 41]. The first research mentioned that AE was applied as an unsupervised deep learning model [51]. The AE consists of an encoder network for transforming the input data into z-latents, and a decoder network utilizes the z-latents as input to reconstruct the output data indicated to the input data. The AE technique has been fashioned to apply in

various applications, such as speech enhancements [52, 53], speech recognition [54, 55], and some research focused on improving the performance of AE [56, 57, 58, 59]. The VQ-VAE has changed the regular AE. The z-latents of AE are continuous variables, but the z-latents of VQ-VAE are discrete variables. The VQ-VAE applies the VQ technique to quantize the continuous variables of the z-latents. Last few years, much research investigated a variety of applications of VQ-VAE. In [60], a comparison of several text compression methods for sequence generation showed that VQ-VAE could be used to compress a text. For the brain recording signal compression [61], the Compressive AutoEncoder (CAE) proposed for deep learning was based on the spike compression model to reduce the data transmission rate. The CAE also used VQ-VAE to compress the brain signal, but the architecture of the encoder and the decoder differed from the original VQ-VAE. In [62], it showed the advantage of the VQ-VAE in decreasing the data size at the communication bottleneck. The VQ-VAE was also introduced to reduce the data usage in controlling statistical speech synthesis [63]. Furthermore, the VQ-VAE was applied in script generation [64] and dialog generation [65].

Nowadays, deep learning has superseded in speech processing fields. The WaveNet [66, 67] model was proposed for text to high-quality speech synthesis. The Generative Adversarial Networks (GAN) [68] was proposed for speech enhancement. The Generative Adversarial Networks is the generative model with high fidelity synthesis, but the model stability in training processes is still a challenging problem compared to the AE. After a few years, GAN became a popular research area, and the trend was about increasing the training stability of the model and the fidelity of the reconstruction of generated data. Some research focused on improving the original GAN model by modifying loss function terms [69, 70] or constraining the discriminator network over the gradient penalties [71]. In [72], the paper concentrated on network architecture modification. In [73], it proposed a cooperation framework between VAE and GAN. The face image reconstruction study indicated that the VAE produced the blurred output image, compared with the GAN and the proposed Variational AutoEncoder Generative Adversarial Network (VAE-GAN).

Furthermore, the GAN technique in the AE was investigated in [56] to match the z-latents of AE to arbitrary distributions. In [74], the AutoEncoder guided GAN technique can improve the performance of supervised AE in calligraphy synthesis problems. The GAN technique that improved the unsupervised AE performance shown in [73] proposed an extension of the VAE model by combining a VAE with GAN that produced fidelity face image

output. The GAN objective distortion major had an advantage compared to the traditional objective distortion such as Mean Absolute Error (L1) and Mean Square Error (L2) to measure the distortion for deep learning methodology.

1.2 Objectives

The objectives of this dissertation are the followings.

(1) To study the advantage of deep learning to improve the reconstruction error of speech spectral envelope quantization.
(2) To study the effect of deep learning architectures for Vector Quantization based on deep learning for speech spectral envelope quantization.
(3) To study the effect of the conventional Vector Quantization techniques based on deep learning for speech spectral envelope quantization.
(4) To study deep learning adversarial techniques for Vector Quantization for speech spectral envelope quantization.
(5) To develop Vector Quantization based on deep learning adversarial technique for speech spectral envelope quantization.

1.3 Structure of this dissertation

The rest of the organization of the dissertation is as follows:

Chapter 2 discusses the literature review of related studies in both conventional speech quantization and end-to-end speech quantization based on deep learning.

Chapter 3 presents the overview of the whole studies in this dissertation.

Chapter 4 describes the first study of the dissertation. The objective is to examine the effect of deep learning architecture on VQ based on fundamental deep learning methods. The conventional VQ, such as K-means or LBG, is a famous speech spectral envelope quantization tool. However, this method limits the reconstruction performance due to using the codebook patterns in the direct domain. On the other hand, the deep learning method can improve reconstruction performance based on the quantization in a perfectly trained z-latent domain. In this chapter, we compare the conventional VQ technique (K-means) with the standard VQ-VAE architecture constructed

from the Convolutional Neural Networks and the Multi-layer Perceptron architecture as the Multi-layer Perceptron Vector Quantized Variational AutoEncoder (MLP-VQ-VAE). The experiments organize four models with different target bitrates. At the end of the chapter, deep learning can improve the reconstruction performance, especially for the proposed MLP-VQ-VAE.

The proposed MLP-VQ-VAE replaces the Convolutional Neural Networks (CNN) with Multilayer Perceptron (MLP) in the architecture of the encoder network and decoder network of VQ-VAE. The CNN made the model which created the z-latents with massive sizes in 3 dimensions and took effect to have a large size of embedding space. The MLP-VQ-VAE can manage the number of z-latent vectors more flexibly than the VQ-VAE and complete the dimensional reduction task. The experiment results evaluated the MLP-VQ-VAE to quantize the spectral envelope parameters of the 48 kHz WORLD vocoder [75] in four target bitrates. As a result, the MLP-VQ-VAE had lower Log Spectral Distortion (LSD) compared to conventional vector quantization and the VQ-VAE and a more reduced representation of z-latents and codebooks or embedding space compared to conventional vector quantization and VQ-VAE.

Chapter 5 is the second study. The objective is to examine the several techniques utilized in the conventional VQ, including the Sub-band VQ and the predictive VQ, for improving the reconstruction error of speech spectral envelope quantization based on deep learning. Four target bitrate VQ-VAE models construct quantizers of the spectral envelope to compare the proposed sub-band Vector Quantized Variational AutoEncoder (sub-band VQ-VAE) and the Predictive Vector Quantized Variational AutoEncoder (Predictive VQ-VAE).

We designed the sub-band VQ-VAE to quantize the speech spectral envelope parameters extracted by the high-quality WORLD vocoder [75]. This vocoder operates with a 48kHz sampling frequency and every 5ms parameter extraction. The parameters include fundamental frequency (F0), spectral envelope parameters (SP), and aperiodic parameters (AP). The WORLD vocoder synthesizes output speech waveforms by using those parameters. We investigated quantization techniques for the SP. In the SP, most human speech information is contained in frequencies below 16 kHz. The frequency above 16kHz has little human speech information. The VQ-VAE uses all the data lengths of SP as input data for quantization. It includes unnecessary speech information in frequencies above 16 kHz. Sub-band coding methods [76, 77] split the data into different frequency bands and compress each band before

transmission. The sub-band VQ-VAE uses this method. It splits SP data into two frequency bands. The first band is the lower frequency band between 0 to16 kHz, and the second is the higher frequency band between 16 and 24 kHz. We assign more bits to the lower band and fewer to the higher frequency band. The performance of the proposed methods is compared with the full-band VQVAE at the same bit rates. In the experiments at four target bitrates, The Sub-band VQ-VAE have lower LSDs at the expense of the larger embedding space size associated with the VQ-VAE.

For the examination of predictive techniques, we proposed the Predictive Vector Quantized-Variational AutoEncoder (Predictive VQ-VAE), an improved version of the conventional VQ-VAE with a prediction mechanism. The evaluation results showed effectiveness for the high-quality 48 kHz WORLD vocoder [75] spectral envelope quantization and the quantization performance by the Log-Spectral Distortion (LSD). Furthermore, the average LSD results indicated that the Predictive VQ-VAE has a lower LSD value than the VQ-VAE for four target bitrates.

In chapter 6, we discuss the fourth and fifth objectives in the dissertation to introduce the advanced deep learning training techniques in VQ-VAE to improve the reconstruction performance. The combination between VQ-VAE and the Generative Adversarial Network was designed to work together in the spectral envelope quantization in four different target bitrates compared to the conventional VQ. The studies investigated the effect of the adversarial loss update on the whole networks of VQ-VAE and only the embedding space of quantization in the VQ-VAE. Based on the results, we proposed the Vector Quantized Variational AutoEncoder Embedding Generative Adversarial Networks as a spectral envelope vector quantization model based on deep learning. The VAEGAN implemented in VQ-VAE, the VQ-VAE-EMGAN, and the VQ-VAE-EMDEC introduced objective distortion measures and training procedures of the GAN technique to replace the conventional distortion measure of VQ-VAE for embedding space learning. The experiments organized four models in each VQ-VAE, the VAEGAN implemented in VQ-VAE, the VQ-VAE-EMGAN, and the VQ-VAE-EMDEC, respectively. The models are trained to quantize the WORLD vocoder's spectral envelope [75]. The spectral envelopes are extracted from the 16 kHz raw speech waveform from the LibriSpeech corpus [78], varied from the 128, 256, 512, and 1024 bits/spectral envelope frame. The quantization performance was evaluated by Log Spectral Distortion (LSD) and the z-latent error (L2). The Perceptual Evaluation Speech of Speech Quality (PESQ), standardized as ITU-T recommendation P.862 [23], is also used to

measure the quality of the reconstructed 16 kHz speech waveform of WORLD vocoder without spectral envelope quantization and with spectral envelope quantization techniques. In the experiments for comparing unquantized z-latents and quantized z-latents for embedding space updating, the results showed that the proposed GAN technique approximated the embedding space better than the Mean Square Error of conventional VQ. The proposed model increased the average PESQ of the waveform by about 0.17 with a reduced average LSD of 0.5 dB with significant results compared to the VQ-VAE and the VAEGAN implemented in VQ-VAE.

Finally, chapter 7 concludes the study and discusses future works opportunities.

Chapter 2

Related Studies

2.1 The speech vocoder

The speech vocoder [24, 25] is the human speech analysis and synthesis system. It encodes human voices into speech parameters and decodes them into the human voice. The speech vocoder was first designed to reduce the data size of the raw speech data in the communication bandwidth and replace the natural carrier sound of the human speech with a synthesized carrier sound at a higher frequency bandwidth. As a result, the speech data could be reproduced more clearly over a long distance since wider frequency band sounds are heard more clearly than narrower ones. The speech vocoder is also helpful in studying the human speech system as a laboratory tool. Text-to-speech tools also incorporate the various vocoders to generate speech sounds.

Linear Prediction Coding (LPC) [28, 29, 30] is the famous speech representation of speech parameters based on human speech production. The fundamental idea of LPC is that human speech signals could be approximated as a linear combination of past speech samples. The essential LPC representation is the all-pole filter model, which approximates human speech. The LPC analysis process consists of spectrum analysis, pitch analysis, amplitude analysis, and the voice or unvoiced decision. This information corresponds to human vocal tract resonance frequencies, pitch repetitions, loudness, and vocal cord vibrations. The synthesis process regenerates speech signals from the parameters in the analysis process.

In the communication system, the raw speech signal is transformed into a digital signal by Pulse Amplitude Moderation (PAM). The digital version of the speech signal is too large to transmit over long-range communication. The vocoder was invented to manage the digital speech data size by analyzing the speech signal into other domains to reduce the data size before applying the transmission system. The well-known vocoder is the Linear Prediction Coding such as (LPC-10) [28, 29, 30] and the Code-Exited Linear Prediction (CELP) [31, 32, 33] in Figure 2.1 (a) and (b). LPC-10 consists of two parts: a feature extraction part at the encoder and a synthesis part at the decoder. On the other hand, CELP has a local decoder at the encoder, which simulates the decoder to find the best codes to minimize the distortion between input and output speech.

Those vocoders are mainly operated in the 8 kHz sampling frequency for telephone line communications. However, the WORLD [75] is a high-quality vocoder that manipulates the 48 kHz raw speech waveform. Figure 2.2 (a) shows the block diagram of the WORLD. First, the analysis system extracts speech parameters from 5ms of input raw speech waveform. The speech parameters consist of the single value fundamental frequency (F0), the vector of the aperiodic parameter (AP), and the vector of spectral envelope parameter (SP). Then, the synthesis system uses these speech parameters to reconstruct the 5ms output speech waveform.

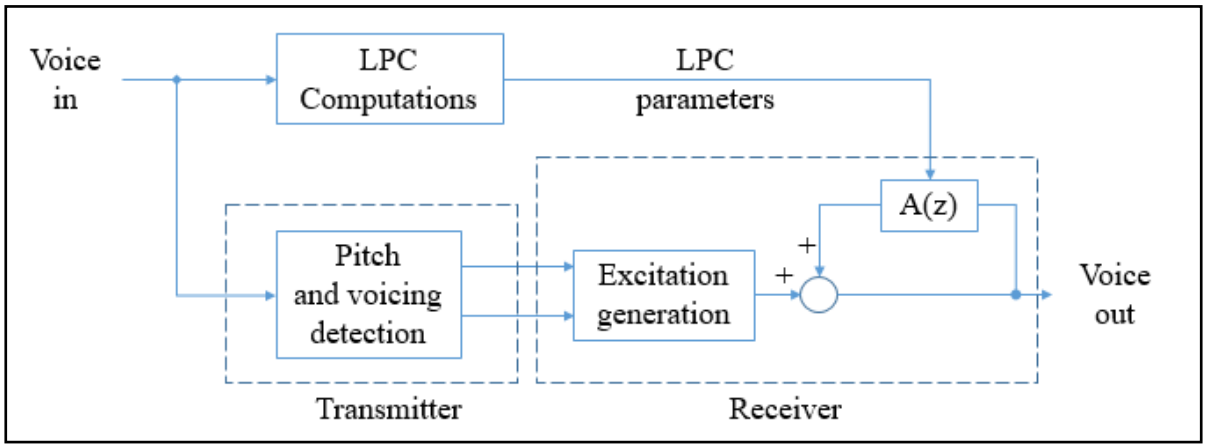

Figure 2.1 (a) Linear Prediction Coding (LPC-10)

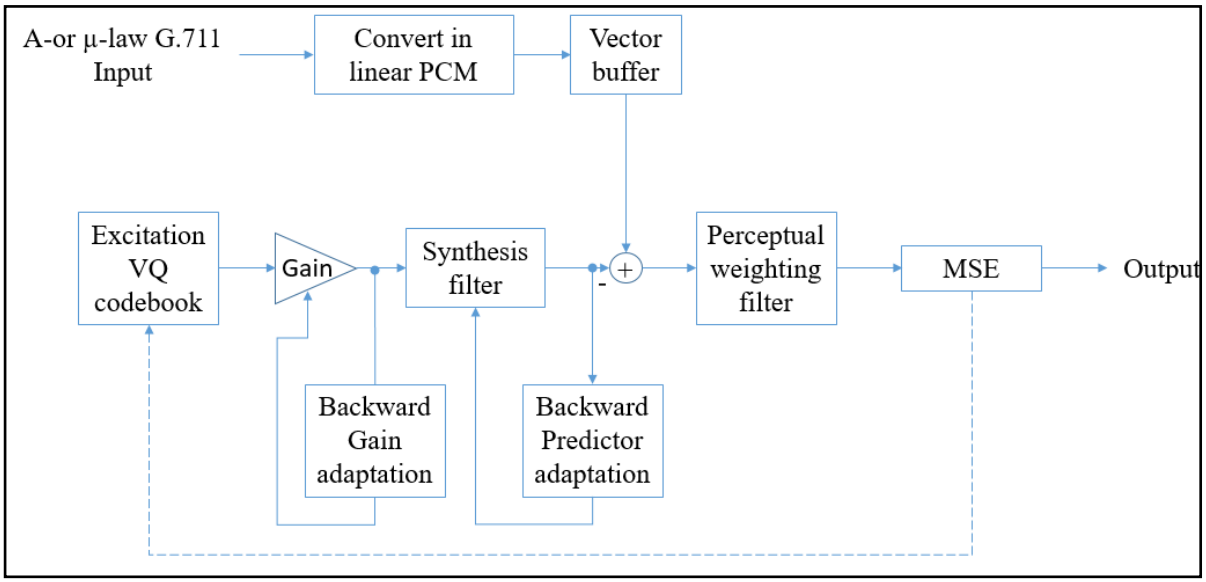

Figure 2.1 (b) Code-Excited Linear Prediction (CELP)

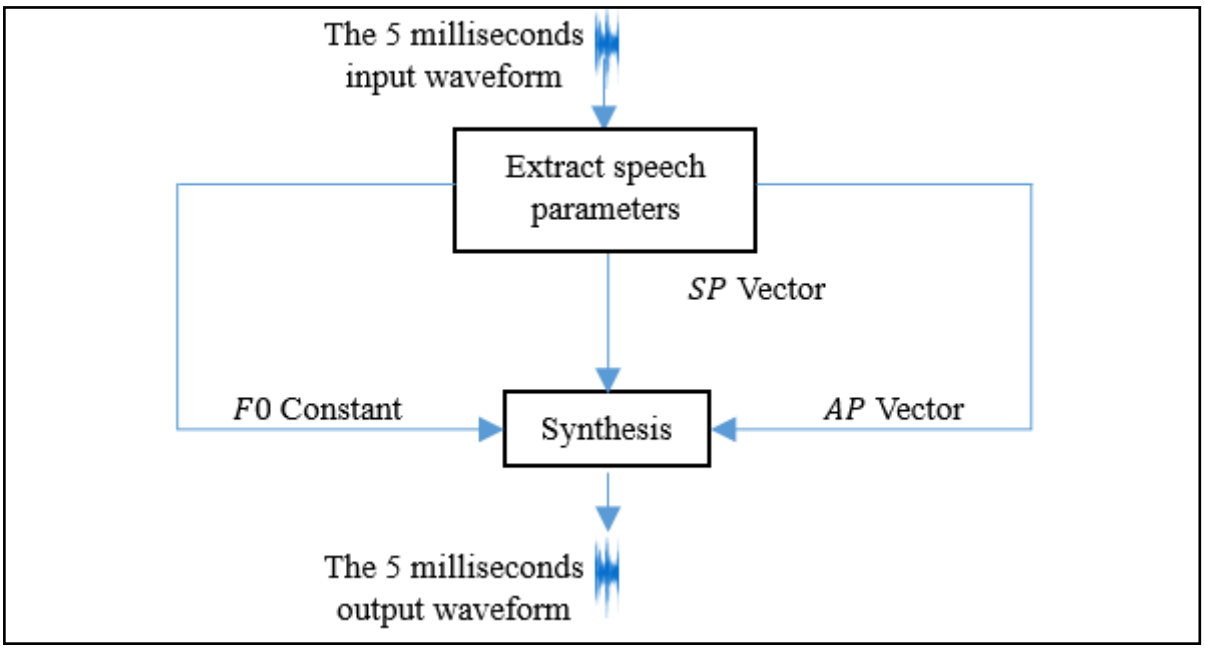

Figure 2.2 (a) The WORLD vocoder

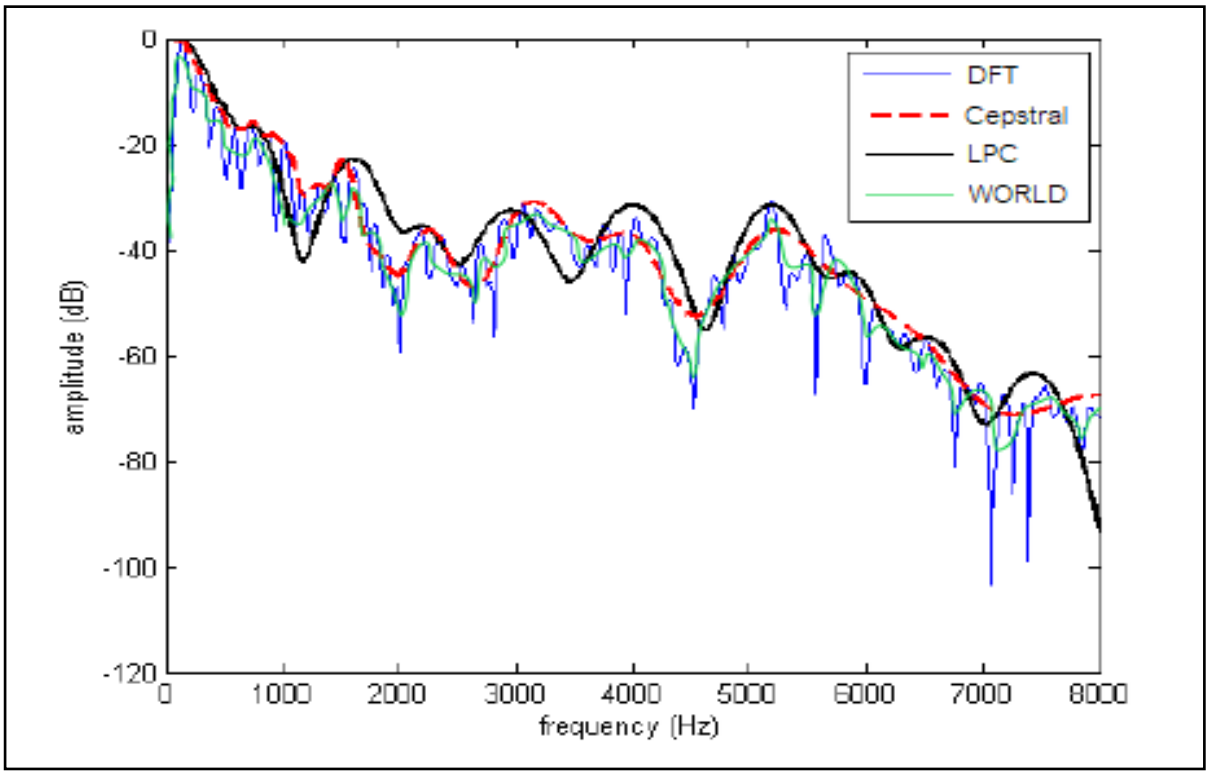

Figure 2.2 (b) The spectral envelope estimation comparison, DFT, Mel-cepstral spectral envelope, LPC-10 spectral envelope, and the WORLD vocoder spectral envelope.

The spectral envelope is the critical speech parameter in speech processing, and Cepstrum and LPC are general methods to represent the spectral envelope with a small number of parameters. However, many methods based on Cepstrum and LPC do not synthesize high-quality natural speech. The difference between the LPC vocoder and the WORLD vocoder is that the LPC vocoder estimates the spectral envelope utilizing the LPC method and represents the LPC coefficients about 10 to 12 coefficients. However, the WORLD vocoder utilizes the full spectral envelope representation from the high-quality spectral envelope estimation method. This point makes the WORLD vocoder synthesize natural speech and show the clear spectral envelope of speech. Figure. 2.2 (b) shows the comparison spectral envelope estimation.

2.2 The Vector quantization

2.2.1 Conventional Vector Quantization

As the VQ method, the popular conventional technique is the LBG, or K-means [40, 41]. The main idea aims to replace the continuous input data with finite discrete representations prepared as a codebook. The codebook is designed with a fixed number of vector patterns. In the encoder, the squared Euclidean distances between the input vector and vector patterns in the codebook are calculated, and the lowest distance vector picks the index number in the codebook. The index number is the code for transmission as the discrete representation. Finally, the decoder obtains the quantized output vector from the vector pattern in the codebook corresponding to the index number. The manner of vector quantization denotes as follows:

$$I = E(x), \tag{2.1}$$
$$\tilde{x} = D(I), \tag{2.2}$$

where x is the input vector, I is the index number, E (.) is the encoder, D (.) is the decoder, and the $\tilde{x}$ is the quantized output vector.

2.2.2 Sub-band Vector Quantization

Figure 2.3 shows the Sub-band Vector Quantization (Sub-band VQ) [76, 77] model. The input vector with a length of x is divided into n sub-vectors, and each sub-vector is quantized by the parallel vector quantizers. The output vector x_q is reconstructed by merging the n quantized sub-band vectors as the quantized data.

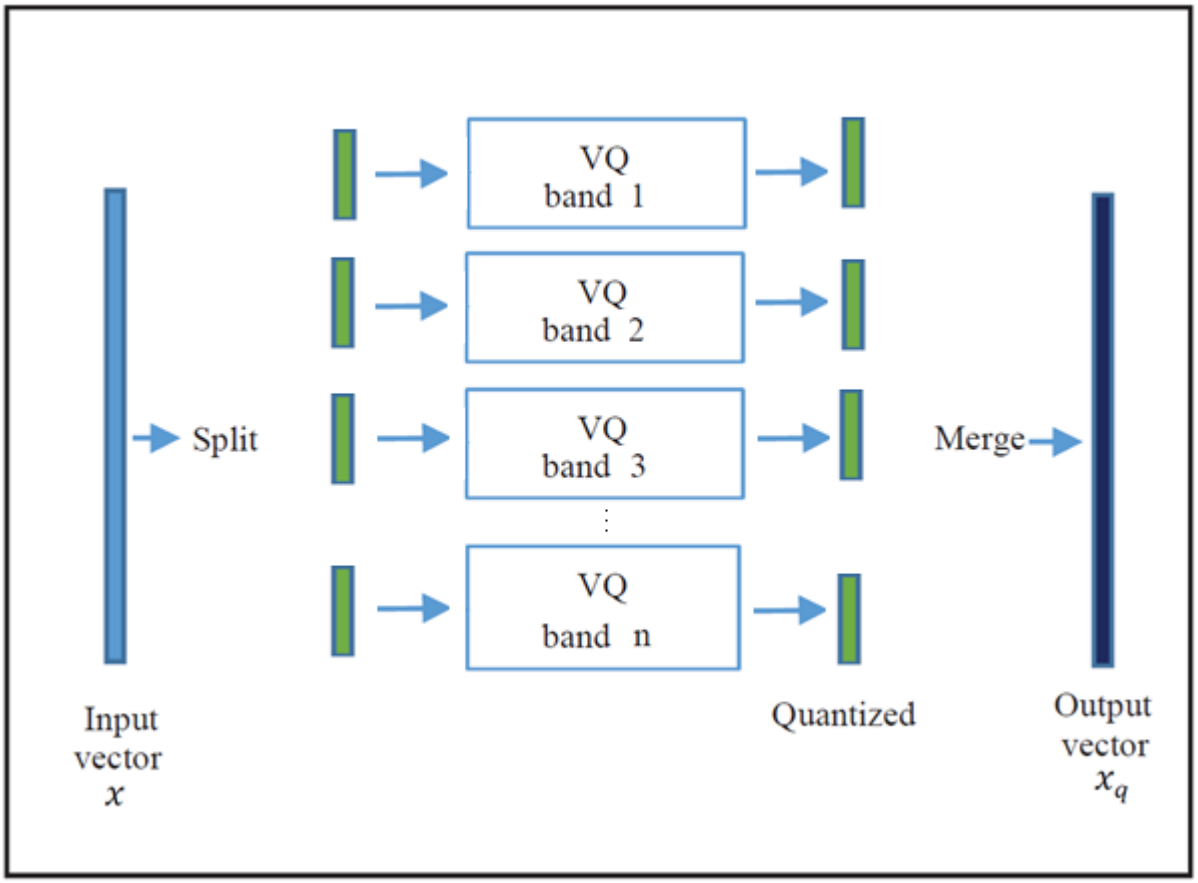

Figure 2.3: The Sub-band Vector Quantization.

2.2.3 Predictive Vector Quantization

The Predictive Vector Quantization (PVQ) [79, 80] was the improved version of conventional vector quantization. Regarding the conventional vector quantization, the current input quantization does not depend on the previous input. The PVQ is designed to use the relationship between the current input and the previous input in the quantization process. The PVQ consists of the encoding and the decoding process, as shown in Figure 2.4. In the encoding process, the input data x is fed into the quantizer to produce both the discrete representation index i and the quantized output data x_q. Then the index i is transmitted to the decoding process. The vector predictor block utilizes the x_q to provide the predicted data $\tilde{x}$ to subtract the following input data x and add with the following output data x_q. In the decoding process, the index i is received from the encoding process, and the quantized output data x_q is reconstructed. The vector predictor outputs the $\tilde{x}$ from the current output x_q, and the $\tilde{x}$ is added to the following output data x_q.

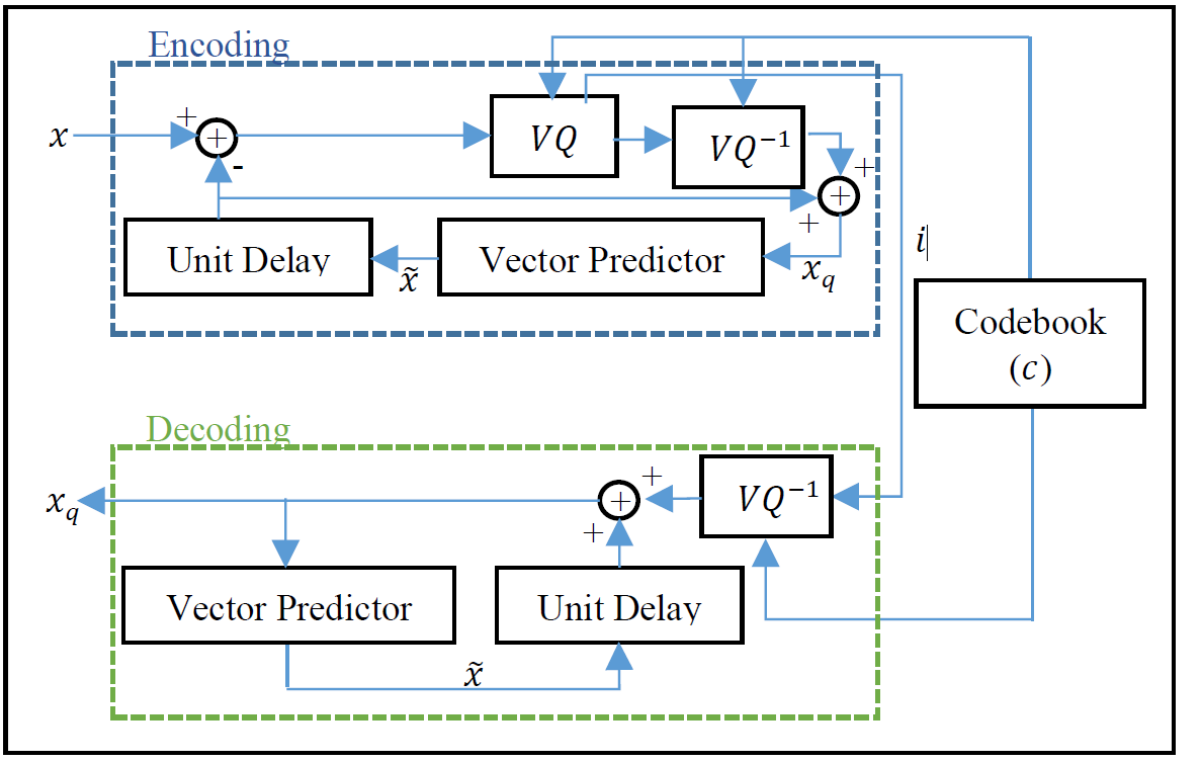

Figure 2.4: The Predictive Vector Quantization.

2.3 Deep learning

2.3.1 The AutoEncoder

The AutoEncoder [51, 52, 53, 54, 55, 56, 57, 58, 59] is a neural network designed for dimensional reduction tasks based on neural networks. The identity function is trained as an unsupervised way to reconstruct the original input while compressing the data in the encoding process and discovering a more efficient and compressed representation. The intermediate representation between the encoder network and the decoder network is called z-latents, which is the reduction of the input data representation. The AutoEncoder block diagram is presented in Figure 2.5. It consists of two networks. The encoder network transforms the original high-dimension input into the z-latent low-dimensional representations, and the decoder network reconstructs the original high-dimensional input data from the z-latent.

The encoder network task is the compressor of input data, and the decoder network is the decompressor for reconstructing the input data from the compressed data called z-latents from the encoder network. The idea of dimensional reduction is similar to the Principal Component Analysis (PCA) but utilizes the neural network.

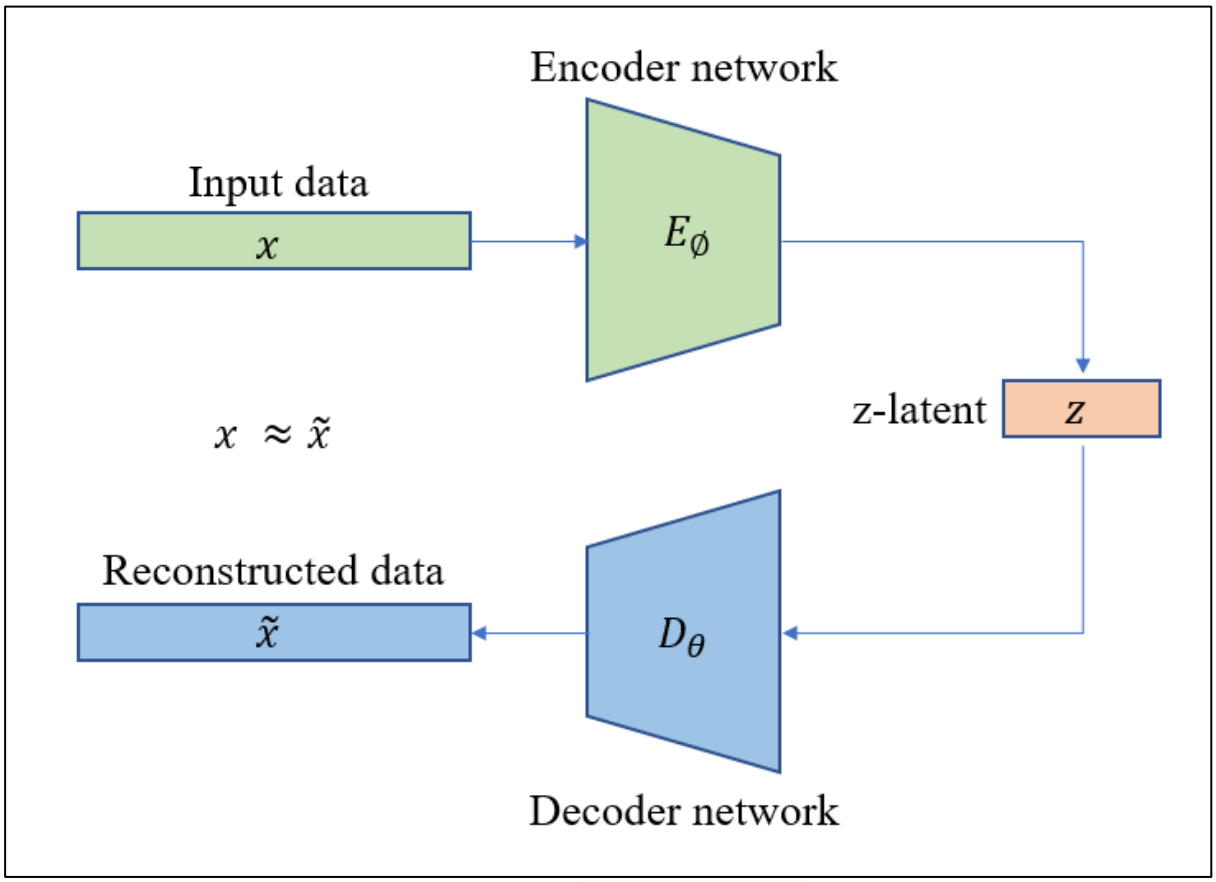

Figure 2.5: AutoEncoder

The AutoEncoder consists of two main networks. The encoder network $E_\phi(.)$ with network parameter ϕ and the decoder network $D_\theta(.)$ with network parameter θ. The intermediate z-latent $z = E_\phi(x)$ is produced from the input data x via the encoder network, and the reconstructed input data is $\tilde{x} = D_\theta\left(E_\phi(x)\right)$ is produced from z received at the decoder network. The encoder network and decoder network parameters (θ, ϕ) are trained to optimize reconstructed data to be the same as the input data by minimizing the loss function in Equation 2.3.

$$L_{AE}(\theta, \phi) = \frac{1}{n}\sum_{i=1}^{n}\left(x_i - D_\theta\left(E_\phi(x_i)\right)\right)^2, \tag{2.3}$$

where $x_i, (i = 1, \ldots, n)$ are the training data, and n is the number of the data.

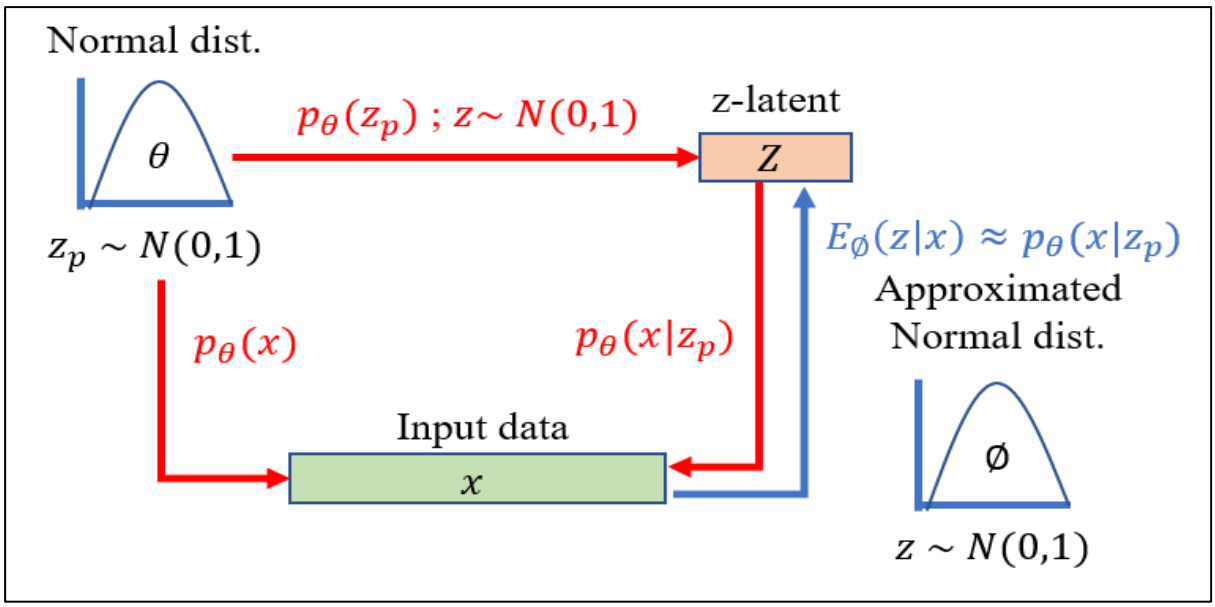

Figure 2.6: VAE z-latent estimation (Normal distribution)

2.3.2 The Variational AutoEncoder

The Variational AutoEncoder (VAE) [48] is designed to produce the z-latent from a distribution. Figure 2.6 shows the VAE z-latent estimation under the condition of normal distributions. The AutoEncoder produces the z-latent without preliminarily hypothesized distributions. On the contrary, the VAE is different. The VAE maps the input vector with the distribution to produce the z-latent with a particular meaning. The label distribution p_θ is parameterized by θ (in the figure, normal distribution). The relationship between the input data x and the z-latent z_p represented by the $p_\theta(z_p)$ is the prior distribution, the $E_\emptyset(z|x)$ is the posterior distribution, and the $p_\theta(x|z_p)$ is the likelihood. The process of reproducing the input data x is started by drawing the z_p from the prior distribution $p_\theta(z_p)$. Then the encoder z-latent outputs z produced from the encoder network $E_\emptyset(z|x)$. The suitable parameter of θ is derived by maximizing the probability of input data x as shown in Equation 2.4.

$$\theta = \max_\theta \sum \log(p_\theta(x)),$$
$$p_\theta(x) = \int p_\theta(x|z_p)p_\theta(z_p)dz, \quad (2.4)$$

where, x is the input data, the $p_\theta(x)$ is the expected function to produce the input data x with parameter θ, the $p_\theta(x|z_p)$ is the likelihood, and the z_p is the z-latent drawn from the normal distribution.

Since it is hard to solve the $p_\theta(x)$ directly, approximation of $E_\phi(z|x)$ is introduced to solve the problem. $p_\theta(x|z_p)$ is the conditional probability with the same function as the decoder network $D_\theta(x|z)$. In the encoder network, the Kullback-Leibler divergence (KL divergence) is used to measure the distance between $E_\phi(z|x)$ and $p_\theta(x|z_p)$. The KL divergence in Equation 2.5 minimizes the function with the encoder parameter ϕ.

$$L_{KL}\left(E_\phi(z|x)||p_\theta(x|z_p)\right) = \log p_\theta(x) + L_{KL}\left(E_\phi(z|x)||p_\theta(z_p)\right) - E_{z\sim E_\phi(z|x)} \log p_\theta(x|z_p). \quad (2.5)$$

From Equation 2.5, the VAE loss function can be defined in Equation 2.6.

$$L_{VAE}(\theta,\phi) = -\log p_\theta(x) + D_{KL}\left(q_\phi(z|x)||p_\theta(x|z_p)\right) = -E_{z\sim q_\phi(z|x)} \log p_\theta(x|z_p) + D_{KL}\left(E_\phi(z|x)||p_\theta(z_p)\right), \quad (2.6)$$

where, the x is the input data, $E_\phi(z|x)$ is the encoder network (approximated normal distribution), $p_\theta(x|z_p)$ is the likelihood, the $p_\theta(x)$ is the expected function to produce the input data x form parameters θ, the z is the encoder network z-latent, and the z_p is the z-latent drawn from the normal distribution.

Figure 2.7 shows the VAE block diagram with the multivariate Gaussian (= normal distribution) assumption. The estimation of the z-latent of encoder network z is expected to be the normal distribution. The reparameterization trick is applied to change the form of the encoder network. The multivariate Gaussian is presented to estimate the normal distribution in Equations 2.7 and 2.8:

$$z \sim E_\phi(z|x) = N(\mu = 0, \sigma^2 = 1), \quad (2.7)$$

$$z = \mu + (\sigma * \epsilon);\ \epsilon \sim N(\mu = 0, \sigma^2 = 1), \quad (2.8)$$

where, the z is the encoder network z-latent, the $E_\phi(z|x)$ is the encoder network, the $N(0,1)$ is the probability of normal distribution, the μ is the mean, σ^2 is the variance, and σ is the standard deviation.

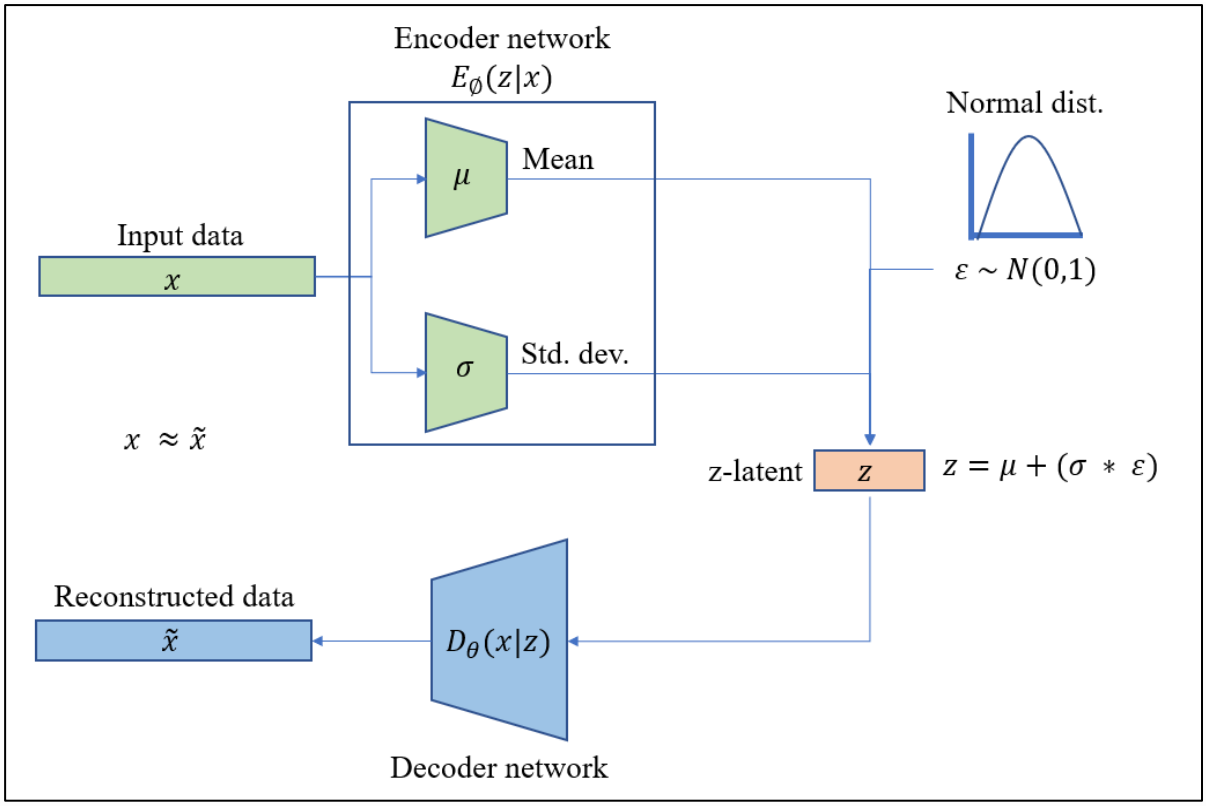

Figure 2.7: VAE with the multivariate Gaussian assumption.

2.3.3 The Vector Quantized Variational Autoencoder

The Vector Quantized Variational AutoEncoder (VQ-VAE) [44,45] is the discrete version of the z-latents in the VAE. The idea of the conventional vector quantization (VQ) is introduced to quantize the continuous z-latents to represent discrete z-latents to apply to specific problems such as classification and digital compression.

Figure 2.8 shows the VQ-VAE block diagram. As conventional VQ uses, the technique transforms continuous z-latent into a discrete format. The embedding space corresponding to the conventional VQ codebook is defined as $e \in R^{K \times D}, i = 1, \ldots, K$, where K is the number of vector patterns in the embedding space, and D is the length of each vector pattern in the embedding space. The z-latent $z_e(x)$ is the output of the encoder network and applied as a reshaping method to change the dimension matching to D of the embedding space. The encoder measures the Euclidean distance between the $z_e(x)$ and the K vector patterns and selects the index of the vector pattern that minimizes the distance in Equation 2.9.

$$z_q(x) = Quantize\big(z_e(x)\big) = e_k\,, \qquad k = \min(\| z_e(x) - e \|^2)\,, \tag{2.9}$$

where the $z_e(x)$ is the encoder network z-latent, the e is the embedding space, and the $z_q(x)$ is the quantized z-latent.

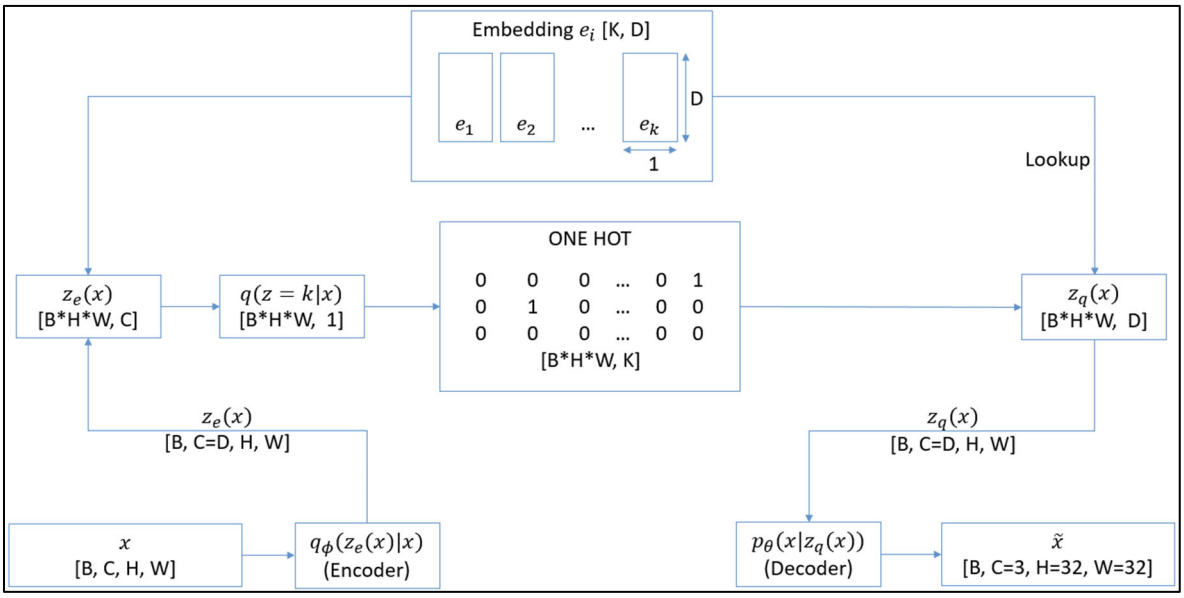

Figure 2.8: Vector Quantized Variational AutoEncoder.

In the backpropagation process, the $\min(.)$ cannot be backpropagated because non-differentiable on the discrete representation. The gradient is calculated from the error between $z_e(x)$ and $z_q(x)$ and added into the loss function of the VQ-VAE in Equation 2.10:

$$L_{vqvae} = -\log p_\theta\left(x\middle|z_q(x)\right) + \|z_e(x) - e\|_2^2 + \beta\|z_e(x) - e\|_2^2, \quad (2.10)$$

where x is the input data, β is the single constant to commit the encoder. $z_e(x)$ is the encoder network z-latent, e is the embedding space, and $z_q(x)$ is the quantized z-latent. In the equation, the first term is the reconstruction loss, which optimizes θ and ϕ of the decoder network $p_\theta\left(x\middle|z_q(x)\right)$ and encoder network $q_\phi(z_e(x)|x)$ parameters. The second term is embedding space loss. Based on the gradients bypass, to obtain the embedding e, the dictionary learning algorithm uses the L_2 error to move the embedding vectors e towards the encoder network z-latent output $z_e(x)$. The third term is commitment loss. Since the embedding space values are dimensionless, they can grow arbitrarily if the embedding's e do not train as fast as the encoder network parameters ϕ. The commitment loss is added to make sure that the encoder network $q_\phi(z_e(x)|x)$ commits to an embedding e.

We propose two types of vector quantization methods in VQ-VAE. The first one is the classic K-means algorithm [40,41]. At the initial state of the model, the ϕ parameters of the encoder network $q_\phi(z_e(x)|x)$, the θ parameters of the decoder network $p_\theta\left(x\middle|z_q(x)\right)$, and the parameters of embedding space are initialized by random noise from the distribution, and the shape is shown in Figure 2.9. The embedding defines the vectors as $e_i, i = 1, \dots, K$, where K is the number vectors, and e_i has the D elements as the length.

Figure 2.10 shows the $z_e(x)$, reshaped $z_e(x)$, and the embedding space. The output of the encoder network reshapes from [H, W, D] into the [H×W, D], in which D is equal to the dimension of the embedding space. Then each vector z_{e_i} ; $i = 1, \dots, n(= H \times W)$ of reshaped $z_e(x)$ finds the distance from each vector e_i ; $i = 1, \dots, K$ in embedding space e:

$$sum\_z_{e_i} = \sum_{j=1}^{D} (z_{e_{ij}})^2;\ i = 1, \dots, n(= H \times W), \quad (2.11)$$

$$sum\_e_i = \sum_{j=1}^{D} (e_{ij})^2;\ i = 1, \dots, K, \quad (2.12)$$

$$Distance_i = sum\_z_{e_i} + sum\_e_i + 2 \times z_e \times e^T. \quad (2.13)$$

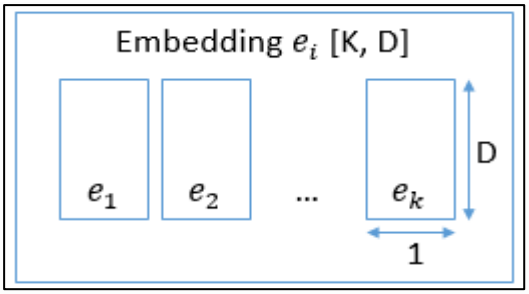

Figure 2.9: Embedding space e.

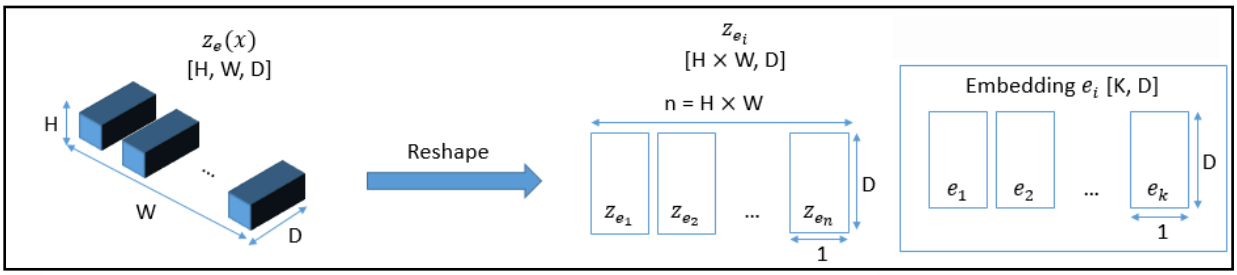

Figure 2.10: $z_e(x)$, reshaped $z_e(x)$, and Embedding space.

Then, each distance $Distance_i$; $i = 1, \dots, n(= H \times W)$ finds the minimum distance value and the indices are encoded into a one-hot vector:

$$indices_i = \underset{j}{\operatorname{argmin}}\left(Distance_{ij}\right); \quad i = 1, 2, \dots, n(= H \times W), and\ j = 1, 2, \dots, K \tag{2.14}$$

$$one\_hot_{ij} = \begin{cases} 1, indices_i = j \\ 0, indices_i \neq j \end{cases}; \quad i = 1, 2, \dots, n(= H \times W), and\ j = 1, 2, \dots, K \tag{2.15}$$

The quantized z-latent $z_q(x)$ is a matrix product of one_hot and embedding e and reshape it to the original shape [H, W, C]:

$$z_q(x) = reshape(one\_hot \times e) \tag{2.16}$$

The last process is loss calculations and performs the loss back propagation to updater VQ-VAE model parameters:

$$Codebook\_loss = MSE\left(z_q(x), z_e(x)\right) \tag{2.17}$$

$$Commitment\_loss = \beta \times MSE\left(z_q(x), z_e(x)\right); \beta = constant \tag{2.18}$$

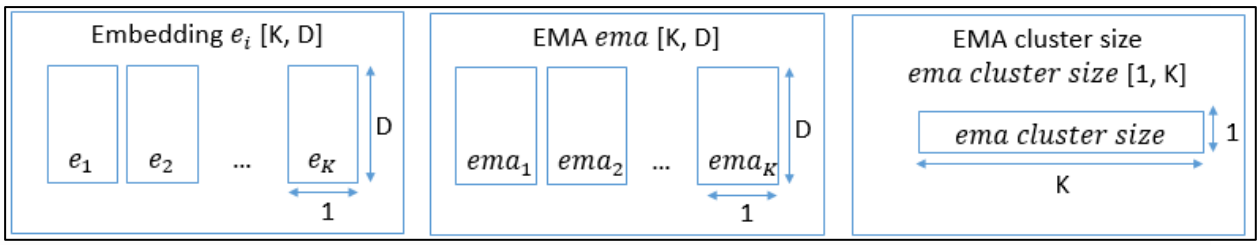

Figure 2.11: Embedding space, EMA, and EMA cluster size.

The second vector quantization method in VQ-VAE is based on the Exponential Moving Average (EMA) method to update the embedding e (codebook). The initial state, the embedding e, the EMA cluster size $ema\_cluster\_size$, and the EMA table ema are defined with the random values from the normal distribution. Figure 2.11 shows the Embedding space, EMA, and EMA cluster size.

The quantization process still uses the same process as the previous K-means technique. It calculates the distance between the output of the encoder $z_e(x)$ and embedding e and returns the indices $indices_i$ corresponding to the minimum distance of each z_{e_i}, then encodes into one_hot. The difference between the first and the second methods is to use the EMA technique to update the embedding e without using the previous codebook loss term in backpropagation to update e.

The EMA process starts with calculating the ema cluster size:

$$ema\_cluster\_size = \frac{ema\_cluster\_size \times decay}{(1-decay) \times \sum_{i=1}^{n=H\times W} one\_hot_i};$$

$$decay = constant\ (0.99). \tag{2.19}$$

Next is to calculate the accumulated vector count N, EMA volume dw, and recalculate the $ema\_cluster\_size$:

$$N = \sum_{i=1}^{K} ema\_cluster\_size_i, \tag{2.20}$$

$$dw = one\_hot^{\,T} \times z_{e_i}, \text{ where } z_{e_i} \text{ refers to reshaped } z_e(x), \tag{2.21}$$

$$ema\_cluster\_size = \frac{ema\_cluster\_size + epsilon}{[N + (K \times epsilon)] \times N};$$

$$epsilon = constant\ (1\times10^{-5}). \tag{2.22}$$

Then, calculate the EMA table ema and update embedding e:

$$ema = ema \times decay + [(1 - decay) \times dw];$$

$$decay = constant\ (0.99) \tag{2.23}$$

$$e = \frac{ema}{ema\_cluster\_size} \tag{2.24}$$

However, the commitment loss still needs to calculate for applying backpropagation technique to update encoder ϕ parameters:

$$commitment\_loss = \beta \times MSE\left(z_q(x), z_e(x)\right) ; \beta = constant \tag{2.25}$$

2.3.4 The Generative Adversarial Networks

The Generative Adversarial Network (GAN) [68, 69, 70, 71] is an unsupervised deep learning technique to generate high-quality output. Figure 2.12 shows the block diagram of GAN. The GAN consists of the generator network to produce the generated data from random noise and the discriminator network to distinguish the real data and the generated data (as fake data) from the generator network. The generator network tries to generate data similar to the real data, and the discriminator networks distinguish between real data and the generated data. The process is a game of two networks, and they try against each other by adopting the generator loss shown in Equation 2.26 to update parameters in the generator network and the discriminator loss in Equation 2.27 for the discriminator network.

$$\min_G L_{GAN}(G) = \mathrm{E}_{z\sim p_z} \log\Big(1 - D\big(G(z)\big)\Big), \tag{2.26}$$

$$\max_D L_{GAN}(D) = \mathrm{E}_{x\sim p_d} \log\big(D(x)\big) + \mathrm{E}_{z\sim p_z} \log\Big(1 - D\big(G(z)\big)\Big), \tag{2.27}$$

where $G(.)$ is the generator network output, $D(.)$ is the discriminator network output, z is random noise, p_z is the data distribution of random noise, x is the real data, and p_d is the distribution of the real data.

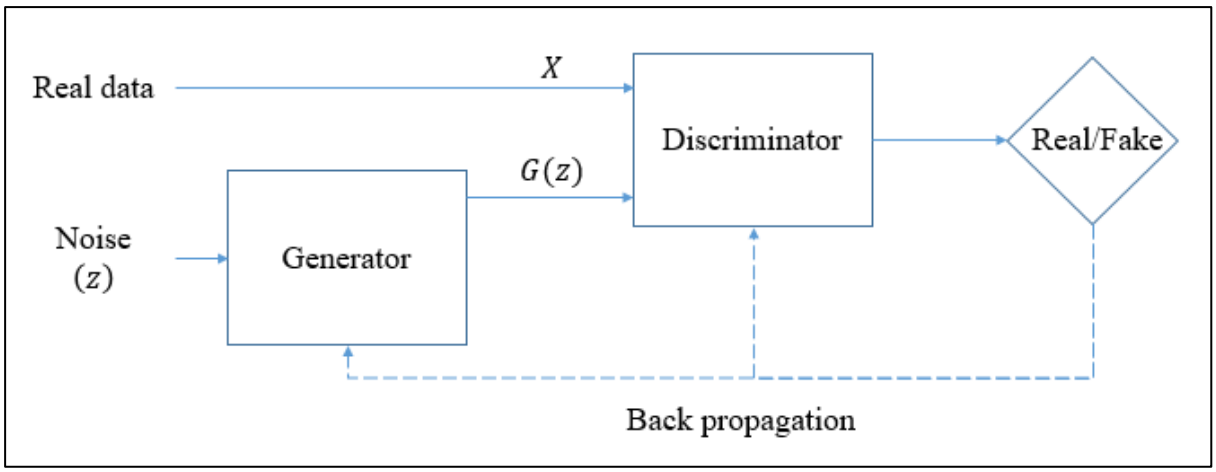

Figure 2.12: The Generative Adversarial Networks

2.4 Perceptual Evaluation of Speech Quality

2.4.1 Subjective Quality Assessments

The Subjective Quality Assessments [81, 82] are the processes to evaluate waveform quality performance by listening and comparing the reference (clean) speech and the transformed (quantized) speech.

The international standard ITU-T Recommendation P.800 defines the Subjective Quality Assessments by the naïve listeners and the headset over one ear. The mean opinion score (MOS) is the average score of the subjective listening tests, scaled from one to five for the worst to the best, as shown in Table 2.1.

2.4.2 Objective Quality Assessments

2.4.2.1 Mean Squared Error

The Mean Squared Error (MSE) [83, 84] is one of the primary objective quality assessments and is applied in many experiments to measure the distortion between the reference (unquantized) and the target (quantized) variables. The formulation of calculating the distortion is presented in Equation 2.28, which calculates the error power of two variable vectors. The MSE value should be minimized for better results (low MSE distortion).

$$L_{MSE} = \frac{1}{n}\sum_{i=0}^{n}(unquantized_i - quantized_i)^2, \qquad (2.28)$$

where *n* is the variable's length, *unquantized* is the unquantized variable, and *quantized* is the quantized one.

Table 2.1
Mean Opinion Score (MOS)

Rating	Label
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

2.4.2.2 L2 Error

The L2 Error is another well-known objective quality assessment method for measuring the distortion between two variables. Equation 2.29 shows the formula for calculating the squared errors of two variables. The L2 is the same trend as the MSE; the value should be minimized for better results (low L2 distortion).

$$L2 = \sum_{i=1}^{n} (unquantized_i - quantized_i)^2, \tag{2.29}$$

where *n* is the length of the variables, *unquantized* is the unquantized variable, and *quantized* is the quantized one.

2.4.2.3 Log Spectral Distortion

The Log Spectral Distortion (LSD) [85, 86] is the distortion measure of objective quality assessments for speech spectral envelopes. The LSD measures the errors of the two spectral envelopes of the root mean square log-spectral distance. The LSD is defined in Equation 2.30. The LSD is also the same trend as the MSE and L2; the value should be minimized for better results (low LSD distortion).

$$LSD_{(dB)} = 10 \times \frac{2}{M} \sqrt{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left(X_{ij} - Y_{ij}\right)^2}, \tag{2.30}$$

where M is the number of log-spectral coefficients frames, N is the vector length of the log-spectral frame, X_{ij} is the original logarithm with base ten unquantized spectral coefficients, and Y_{ij} is the quantized logarithm with base ten spectral coefficients. In this research, we use the spectral coefficients of the WORLD.

2.4.2.4 Perceptual Evaluation of Speech Quality (PESQ)

The PESQ [87] is the ITU-T Recommendation P.862. The PESQ is a standard objective quality assessment measure for calculating the waveform distortion at an 8 kHz sampling rate for narrow-band and 16 kHz for wide-band. The PESQ measures the distortion between two speech waveforms. The objective score was developed to estimate the traditional MOS method of subjective quality assessment. The score varies from the lowest to the highest (-0.5 to 4.5). The PESQ MOS score output a higher score when the distortion between two waveforms is lower. The maximization of the PESQ MOS score helps improve the waveform's subjective quality.

Chapter 3

Overview of speech spectral envelope quantization based on deep learning

3.1 Overview

This chapter discusses the general idea of the development flow for speech spectral envelope quantization. We explain each study's concept and how it relates to the final gold of speech spectral envelope quantization based on deep learning.

The speech parameters produced by the WORLD vocoder [75] analysis consist of the Fundamental Frequency (F0), the Aperiodicity Parameter (AP), and the Spectral Envelope Parameter (SP). Those parameters reproduce the raw input speech waveform at the vocoder's output. The most critical parameter for synthesizing high-quality output speech waveform is the SP compared to the F0 and AP. The F0 is the constant that represents the Fundamental Frequency of the input speech frame (5 ms raw speech input data). The Aperiodicity is a vector of indices in the frequency domain. The vector shows either voice frequency parts or unvoiced/noisy frequency components. The Spectral Envelope is the vector of speech frequency envelope estimation of the raw speech input. The SP is the most sensitive parameter for noise compared to F0 and AP. Suppose the SP got the same distortion with the F0 and AP. The distorted SP will reproduce the bad quality output speech waveform. On the other hand, if the F0 and the AP got some distortion, the synthesized output speech waveform quality is not affected so badly. The bit allocation is also an important parameter when designing the quantizer.

This dissertation studies WORLD Spectral Envelope Quantization by applying the conventional and recent quantization techniques based on Deep learning. Figure 3.1 shows the overall system of the WORLD spectral envelope quantization process. First, the 5 ms input is analyzed, the SP vector is extracted, and F0 and Ap parameters are extracted. Secondly, the SP vector is logarithmically compressed and normalized between Min-Max limits, obtaining SP_{norm} vector. Then, proposed quantization methods are applied to the SP_{norm}, and SP_{norm_q} is obtained. The WORLD synthesizes the output speech by using the SP_{norm_q} vector, the F0 constant, and the AP vector.

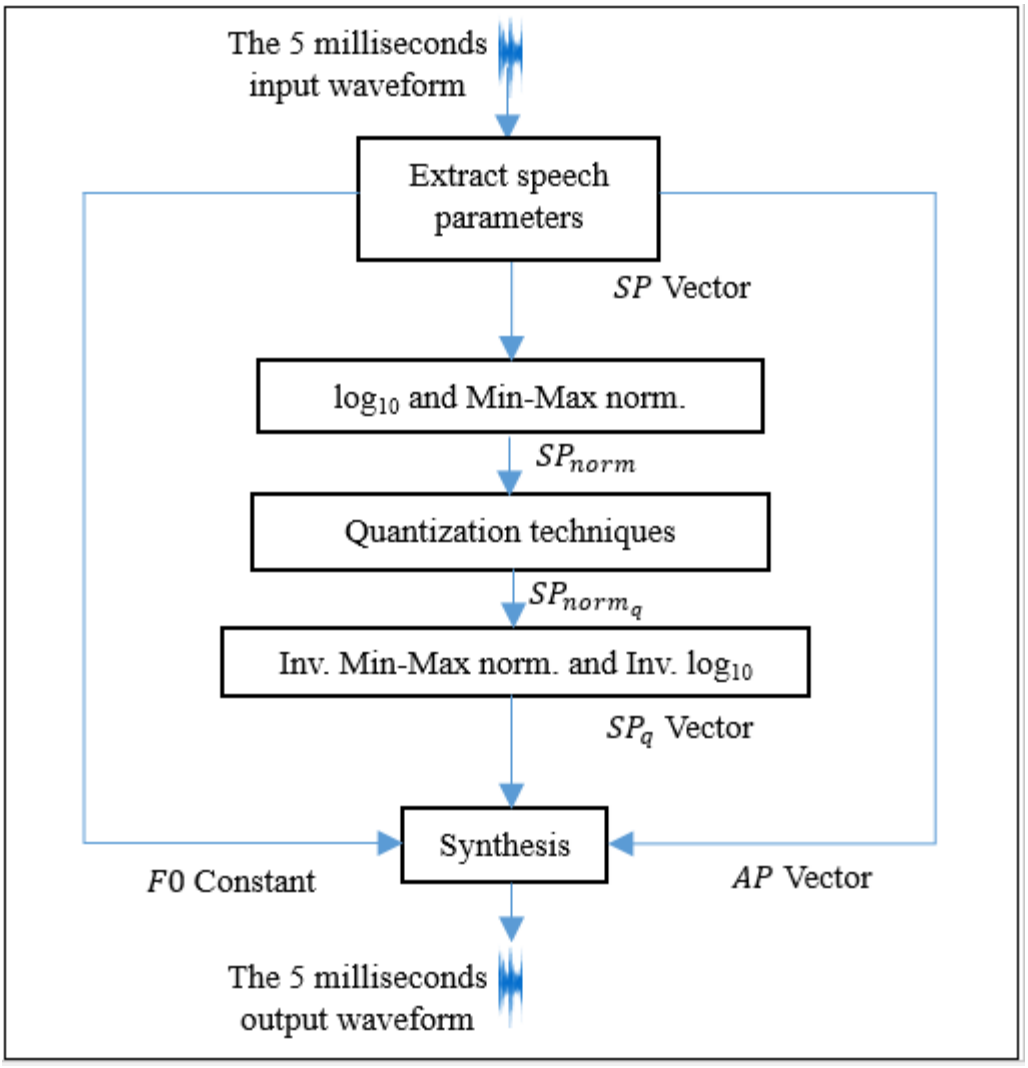

Figure 3.1: The WORLD vocoder Spectral Envelope Quantization

3.2 Scope of the Spectral Envelope Quantization

In this dissertation, the conventional Vector Quantization (K-means) [40, 41] is applied to quantize the WORLD spectral envelope compared with deep learning techniques. The database of the raw speech waveform utilized in this dissertation is the VCTK corpus (48 kHz sampling frequency) and the LibriSpeech corpus (16 kHz sampling frequency) to study the effect of deep learning quantization techniques.

In Chapter 4, three Vector Quantization techniques are investigated for the performance to quantize the WORLD spectral envelopes of 48 kHz raw speech waveforms. The VQ techniques include the K-mean, the Vector Quantized Variational AutoEncoder (VQ-VAE), and the proposed Multi-

Layers Perceptron (MLP-VQ-VAE). The proposed MLP-VQ-VAE takes advantage of the conventional technique (K-means) and the Vector Quantization based on deep learning (VQ-VAE) in terms of the reconstruction error.

In Chapter 5, the fundamental two conventional Vector Quantization methods such as the Sub-band Vector Quantization [76, 77], the Predictive Vector Quantization [79, 80] are investigated in the Vector Quantization based on the frameworks of deep learning called Vector Quantized Variation AutoEncoder (VQ-VAE) by utilizing the 48 kHz raw speech waveforms, the proposed Sub-band VQ-VAE, and the Predictive VQ-VAE. In the experiments, the conventional VQ method can boost the reconstruction performance of the VQ-VAE.

In Chapter 6, the Adversarial Deep learning Techniques such as the Generative Adversarial Technique [68] are combined with the VQ-VAE to reduce the z-latent errors for the 16 kHz raw speech waveform. The Adversarial Loss terms and the Adversarial Deep learning Technique can increase the z-latent reconstruction performance compared to the conventional Euclidian distance error of VQ-VAE.

Chapter 7 is the conclusion of the dissertation. The whole study is summarized into the point and mentions the advantages and disadvantages of the VQ based on deep learning. The end of the chapter also presents the future work in the last session.

Chapter 4

The vector quantization based on deep learning for speech spectral envelope quantization

4.1 Overview

This section aims to study the effect of deep learning architecture on VQ. The conventional VQ technique (K-means) [40, 41] constructed to quantize the spectral envelope in four different target bitrates are compared to the standard VQ-VAE [44, 45] that architecture is constructed from the Convolutional Neural Networks and the Multi-layer Perceptron architecture as the Multi-layer Perceptron Vector Quantized Variational AutoEncoder (MLP-VQ-VAE).

Figure 4.1 shows an overview of the spectral envelope quantization. The MLP-VQ-VAE is proposed to replace the Convolutional Neural Networks (CNN) [88] with Multilayer Perceptron (MLP) [89] in the architecture of the encoder network and decoder network of VQ-VAE. The CNN makes a model that creates the z-latents with massive sizes in three dimensions and takes effect to have a large size of embedding space. The MLP-VQ-VAE can manage the number of z-latent vectors more flexibly than the VQ-VAE and complete the dimensional reduction task.

The experiment results evaluated the MLP-VQ-VAE to quantize the spectral envelope parameters of the 48 kHz WORLD vocoder [75] in four target bitrates. It showed that the MLP-VQ-VAE had lower Log Spectral Distortion (LSD) compared to conventional vector quantization and the VQ-VAE and had a smaller representation of z-latents and codebook or embedding space compared to conventional vector quantization and VQ-VAE.

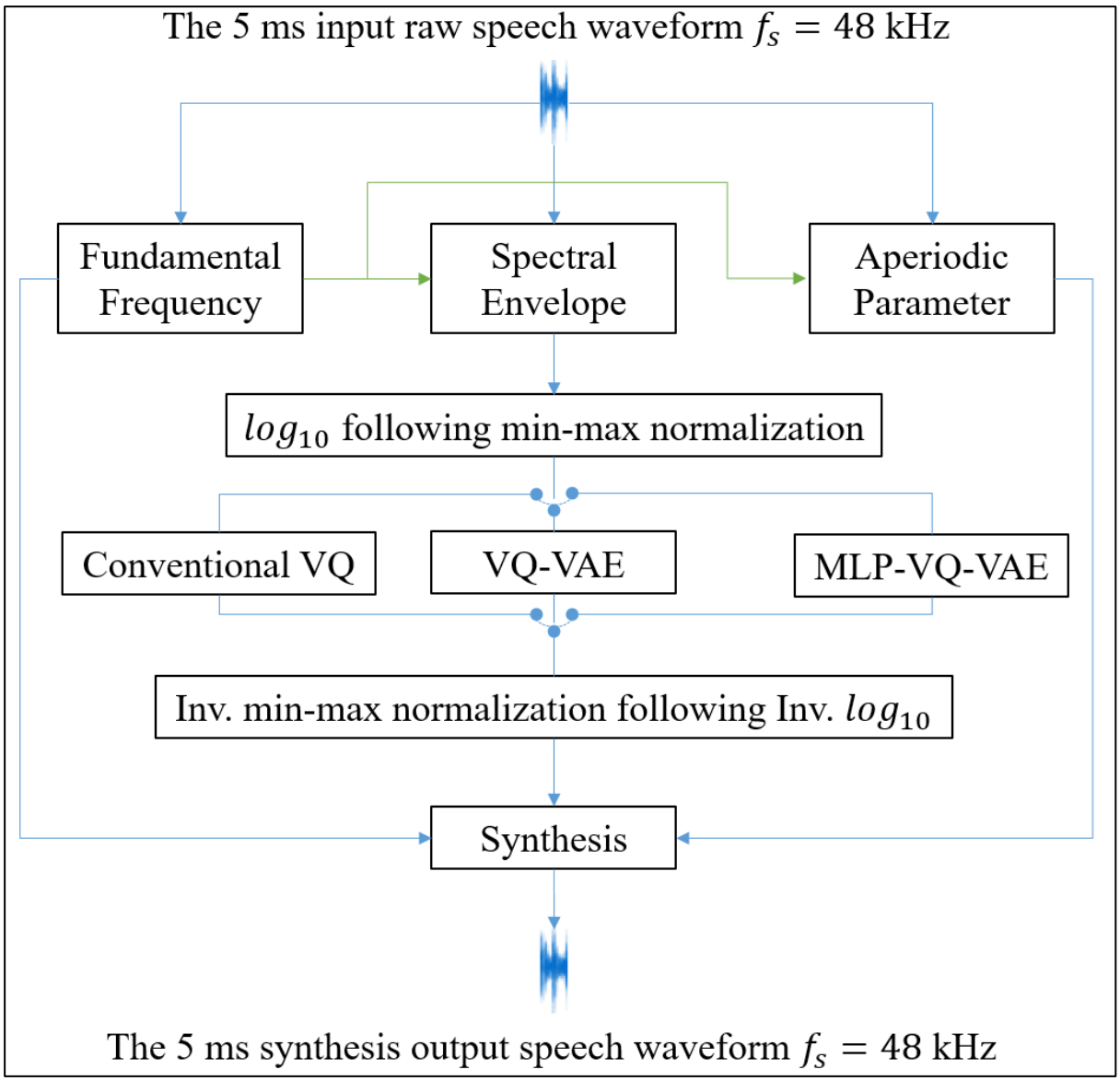

Figure 4.1: Overview of the well-known conventional vector quantization compared to the vector quantization based on deep learning for speech spectral envelope quantization.

4.2 Methodology

4.2.1 Conventional Vector Quantization

The LBG or K-means [40, 41] is used as popular conventional VQ methods. the main idea aims to replace the continuous input data with finite discrete representations. The codebook is designed with fixed number of vector patterns. In the encoder, the squared Euclidean distance between the input vector and the vector patterns in the codebook is calculated, and the index number is picked by the lowest distance vector in the codebook. The index

number is the discrete representation. In the decoder, the quantized output vector is constructed from the vector pattern in the codebook directed by the index number. The manner of vector quantization denotes as follows:

$$I = E(x) \quad (4.1)$$
$$\tilde{x} = D(I) \quad (4.2)$$

where x is the input vector, I is the index number, E (.) is the encoder, D (.) is the decoder, and the $\tilde{x}$ is the quantized output vector.

4.2.2 Vector Quantized Variational AutoEncoder (VQ-VAE)

The Vector Quantized-Variational AutoEncoder (VQ-VAE) [44, 45] is an end-to-end vector quantization method based on deep learning, inspired by conventional vector quantization. The encoder network is implemented in the model with two stride convolutional layers (stride by 2) connected to two residual blocks and one convolutional layer to prepare the z-latent shape (stride by 1) and the fixed embedding space utilized for z-latent quantization. The decoder network is a counterpart of the encoder network, in which one transposed convolutional layer (stride by 1) is used for recovering the suitable shape for the two residual blocks, followed by two transposed convolutional layers.

The input data x are fed into the encoder network to produce the z-latent $Z_e(x)$. The reshape technique is applied to $Z_e(x)$, and the reshaped z-latent is sub-vectors with suitable shape for quantization that corresponds to vector patterns in embedding space e. The vector quantization is applied to each sub-vector of reshaped z-latent by calculating the squared Euclidean distance and returning a set of one-hot vectors for discrete representation. From the set of one-hot vectors, the quantized version of reshaped z-latent is organized by selecting the vector pattern that corresponds to the set of one-hot vectors from e. The quantized version of reshaped z-latent is obtained by reshaping the original shape as the $Z_e(x)$ to represent the quantized z-latent $Z_q(x)$. The decoder network uses the $Z_q(x)$ as input to produce the output data $\tilde{x}$ that indicates the input data x. The overall training loss consists of three terms. The first term is negative log-likelihood for optimizing the encoder and the decoder networks about reconstruction error. The second term is the least-squares error of changes of vector patterns in e toward the reshaped z-latent. The last term is the commitment loss to make sure that the e can be trained as fast as the encoder network. The loss is described as follows:

$$L = -\log(p(x|Z_q(x)) + \|sg[Z_e(x)] - e\|_2^2 + \beta\|Z_e(x) - sg(e)\|_2^2 \quad (4.3)$$

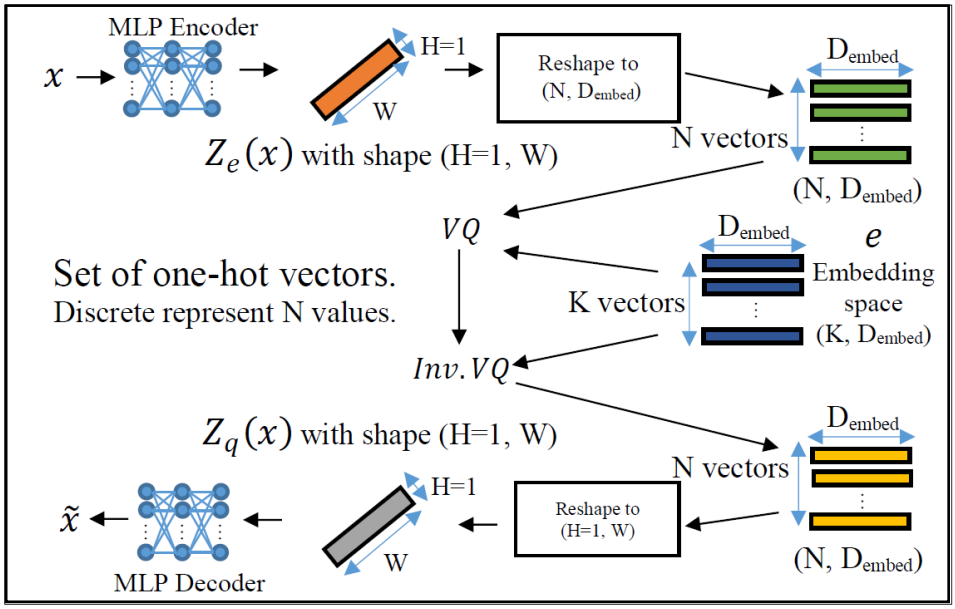

Figure 4.2: The block diagram of MLP-VQ-VAE.

where x is the input data, the $Z_e(x)$ is the z-latent, the e is embedding space, the $Z_q(x)$ is the quantized z-latent, the sg (.) is the stop gradient operator, and β is the hyper-parameter.

4.2.3 Multilayer Perceptron Vector Quantized Variational AutoEncoder (MLP-VQ-VAE)

We propose the Multilayer Perceptron Vector Quantized Variational AutoEncoder (MLP-VQ-VAE) to improve the flexibility to control the z-latent size and entire dimensionality reduction. The VQ-VAE encoder network transforms the input data into a z-latent. The stride and the filter parameters control the size of the z-latents. However, since the encoder network cannot complete the dimensional reduction task, the z-latent is still vast, and hard to control the size. The MLP-VQ-VAE replaces the convolutional neural network with the simple Multilayer Perceptron (MLP). As a result, the MLP can efficiently operate the dimensional reduction of the input data and has the flexibility to control the z-latent size.

As shown in Figure 4.2, the model consists of the encoder and decoder networks formed with MLP to cooperate with the embedding space (e) related to quantization. Suppose x is the input data. The encoder network transforms x into the z-latent vector $Z_e(x)$ that is a smaller representation with the shape (H = 1, W), where H is the height dimension, and W is the width dimension. The fixed embedding space is designed with the shape (K, D_{embed}), where K

is the number of vector patterns, and D_{embed} is the length of the vector. The $Z_e(x)$ is reshaped into N sub-vectors with the length of D_{embed}. The vector quantization is applied to each sub-vector of reshaped z-latent to obtain the set of one-hot vectors. The quantization of reshaped z-latent sub-vectors is constructed by choosing the vector pattern that corresponds to the one-hot vectors, and it reshapes back to (H = 1, W) to represent the quantized z-latent $Z_q(x)$. The decoder network uses the $Z_q(x)$ to produce the output data $\tilde{x}$ that indicates to x. The training loss function in the practice of the MLP-VQ-VAE also uses the same loss of the VQ-VAE in Equation (3) for model optimization.

4.3 Experiments and Results

4.3.1 WORLD vocoder spectral envelope quantization

The WORLD vocoder [57] is the high-quality vocoder that manipulates the 48 kHz raw speech waveform. The analysis system extracts speech parameters from 5 milliseconds of input raw speech waveform. The speech parameters consist of the single value fundamental frequency (F0), the vector of the aperiodic parameter (AP) with the length of 1025, and the vector of the spectral envelope parameter (SP) with the length of 1025. The synthesis system uses speech parameters to reconstruct the five milliseconds output speech waveform. The SP parameter is a highly complex structure and necessary to synthesize speech waveform. The proposed MLP-VQ-VAE was applied to quantize the SP parameter and compared with the conventional vector quantization and VQ-VAE.

The WORLD analysis and synthesis processes in the experiment were as follows: The five milliseconds input raw speech waveform was applied to extract speech parameters. The constant of F0 and the AP vector were sent to the synthesis part without applying the quantization. The SP vector was applied to the logarithm base 10, followed by the min-max normalization to scale values between 0 and 1, and vector quantization techniques were utilized to quantize. The quantized SP vector was applied to the inverse min-max normalization and the inverse logarithm base 10 to transform data into the original values of the quantized spectral envelope parameter. The synthesis system used the F0 constant, AP vector, and quantized SP vector to reconstruct the five milliseconds output speech waveform.

Table 4.1
The implementation of four conventional vector quantization models

Vector Quantization techniques	SP vector	Reshaped vectors (N, D_{embed})	Coodbook (K, D_{embed})	$\frac{Bits}{SP\ vector}$
Conventional Vector Quantization	(1, 1025)	(5, 205)	(512, 205)	45
		(25, 41)	(512, 41)	225
		(41, 25)	(512, 25)	369
		(205, 5)	(512, 5)	1845

4.3.2 Raw speech waveform database

We used the CSTR VCTK corpus [90] as the raw speech waveform database. The speech database is 16 bits with 48 kHz as sampling frequency, recorded from 109 English native speakers with about 400 sentences with various accents. To construct the SP vector database, the WORLD vocoder appropriately extracted the spectral envelope parameters from every five milliseconds of the waveforms. Then, in the training process of vector quantization techniques, the SP vector database was processed with the logarithmic base ten and min-max normalization to transform the values into a scale between 0 to 1.

4.3.3 The comparison of MLP-VQ-VAE and conventional vector quantization for spectral envelope quantization

The experiment investigated the MLP-VQ-VAE and the conventional vector quantization performance for quantization of the SP vector. The SP vector length was 1025. The codebook for conventional vector quantization and the embedding space for MLP-VQ-VAE was fixed with the same number of K = 512 vector patterns or 9 bits quantization.

We implemented four models to operate in four different bitrates as the conventional vector quantization. Because of the limitation of the reshaping method, it affected finding the number of N sub-vectors. The four shapes of reshaped vectors were selected, and the implementation detail for the four models was shown in Table 4.1. The first model reshaped the SP vector into shape (N = 5, D_{embed}= 205) to represent the reshaped vector for quantization, where N is sub-vectors to quantize, and D_{embed} was the length of the vector corresponding to the length of vector patterns in the codebook. The VQ process assigned each sub-vector with the nearest vector patterns in the codebook; the first model is 45 bits/SP vector. The reshaped vectors are (N = 25, D_{embed} =

41), (N = 41, D_{embed} = 25), (N = 205, D_{embed} = 5), the bitrates became 225 bits/SP vector for the second model, 369, and 1845 bits/SP vector for the third, and the fourth model, respectively.

Figure 4.3 shows the quantization process of conventional VQ. The input SP vector was reshaped into the compatible shape for quantization, related to the designed Codebook vector patterns. The VQ process was applied to quantize each of the reshaped vectors to transform the continuous vector into a discrete representation. The inverse VQ received the discrete representation and reconstructed the quantized version of the reshaped vectors by selecting the corresponding vector patterns in the codebook that matched the discrete representation. At the end of the quantization process, the quantized reshaped vector was transformed into the quantized SP vector by the applied reshaping method.

Figure 4.4 shows the Conventional VQ Training process. The training of four conventional vector quantization models utilized the maximum number of iterations of the k-means algorithm set for 300 iterations and relative tolerance with regards to inertia to declare convergence set to 0.0001.

In the training process, the input SP vector calculated the distance (Euclidean distance) with all K vector patterns in the codebook with Equation 4.4:

$$d_i(p, q_i) = \sqrt{(p - q_i)^2} \quad ; \quad i = 1,2, \dots, K, \tag{4.4}$$

where p was the SP vector and q_i was the vector pattern in the codebook. K was the number of vector patterns in the codebook.

The discrete representation of the input SP vector was represented with the index number I of the minimum distance d_i, calculated from Equation 4.5:

$$\min_{q_i \in K} d_i(p, q_i), \tag{4.5}$$

where p was the SP vector and q_i was the vector pattern in the codebook. min(.) was the minimum function.

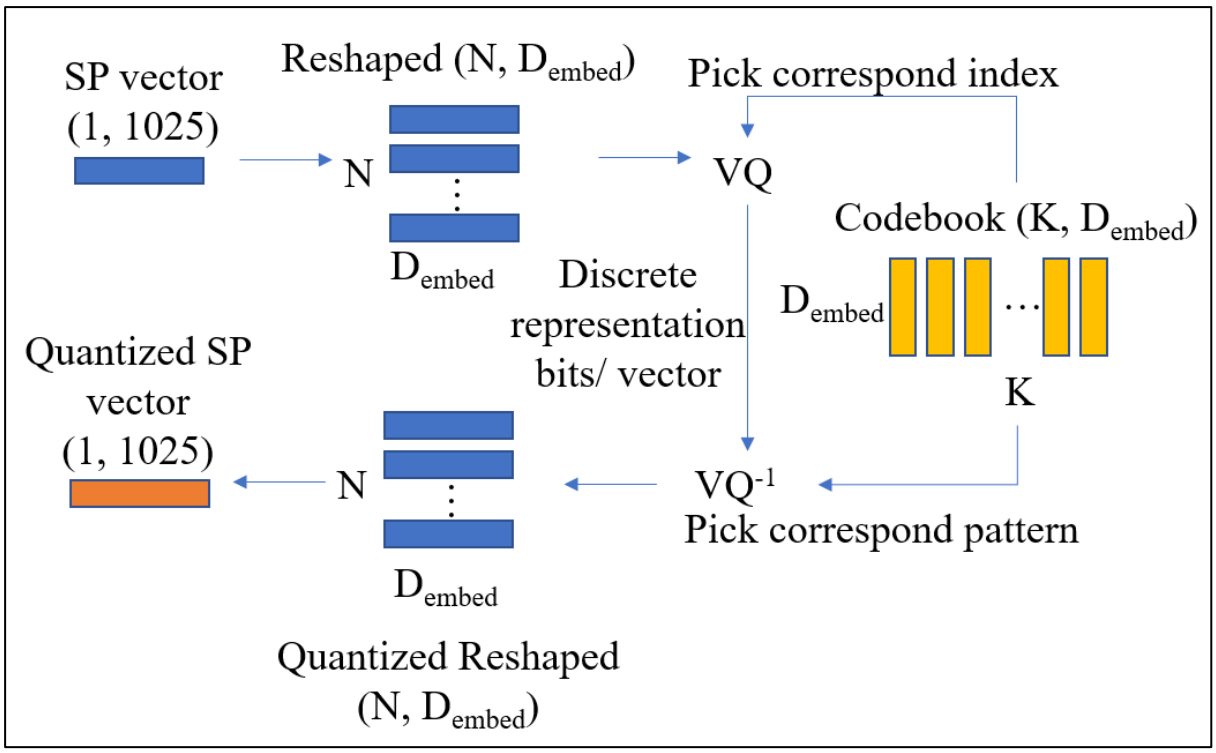

Figure 4.3: The Conventional VQ quantization diagram.

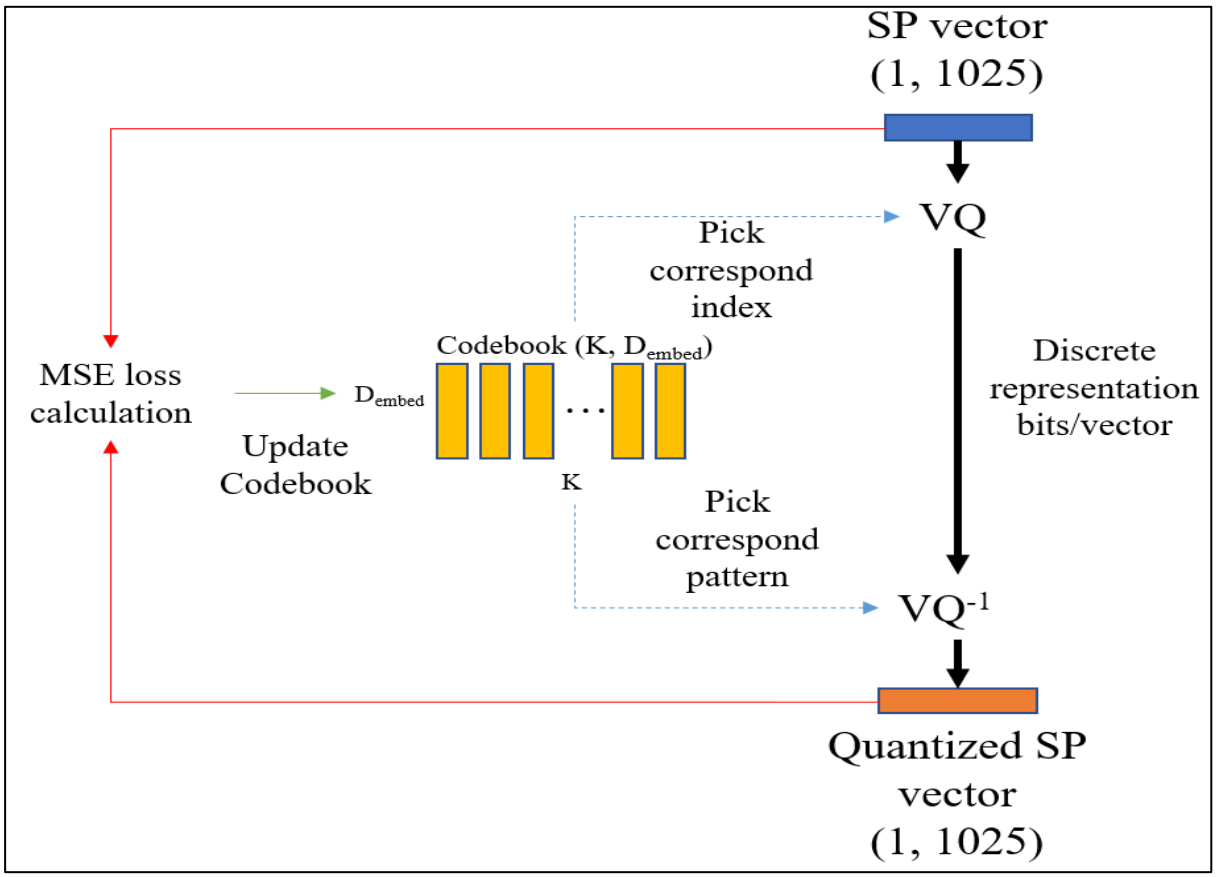

Figure 4.4: The Conventional VQ Training process.

The final training process updated the vector pattern in the codebook corresponding to the input SP vector with Equation 4.6:

$$k_i = \frac{1}{|S_i|} \sum_{p_j \in S_i}^{M} p_j \; ; i = 1,2, \dots, K \; and \; j = 1,2, \dots, M, \tag{4.6}$$

where S_i is the input SP vector cluster indexed as i. K is the number of vector patterns (centroids) in the codebook. M is the number of input SP vectors in each cluster. K is the number of vector patterns (centroids) in the codebook. p_j $(j = 1, \dots, M)$ is the SP vector in the cluster S_i, and k_i $(i = 1, \dots, K)$ is the i-th vector pattern (centroid) with a minimum distance from SP vectors in the cluster S_i.

The MLP-VQ-VAE architecture was constructed with Multilayer Perceptron networks. The encoder network consists of three layers. The input layer assembled with 1025 neurons, 1025 bias, and Rectified Linear Unit (ReLU) was used as the activation function. The hidden layer was assembled with 900 neurons, 900 biases, and ReLU. The coding layer was flexibly designed with a specific number of neurons with the same number of biases. The decoder network was a counterpart of the encoder, and the input layer used the same structure as the coding layer of the encoder network. The hidden layer was also the same structure as the hidden layer of the encoder network. The output layer was built with the same structure as the input layer of the encoder network, but Sigmoid was implemented as the activation function.

Table 4.2 shows the implementation conditions of the four models of MLP-VQ-VAE, designed with the four target bitrates to compare with the conventional vector quantization. The number of neurons and bias were the same. The ReLU was applied to the input layer and the hidden layer as the activation function, while it was not to the coding layer. The encoder networks were 1025-900-615, 1025- 900-625, 1025-900-615, and 1025-900-615 for the first, second, third, and fourth models, respectively. The z-latent vector sizes were (1, 615), (1, 625), (1, 615), (1, 615), the reshaped z-latents were (N = 5, D_{embed} = 123), (N = 25, D_{embed} = 25), (N = 41, D_{embed} = 15), and (N = 205, D_{embed}= 3), for the first, second, third, and the fourth models, respectively. The number of N sub-vectors was set as the same as the four conventional vector quantization models, and D_{embed} was the length corresponding to the length of vector patterns in the embedding space. The bitrates became 45, 225, 369, and 1845 bits/SP vector, respectively.

Table 4.2
The implementation of four MLP-VQ-VAE models for comparison with conventional vector quantization

Vector Quantization techniques	SP vector	z-latent vector	Reshaped z-latent (N, D_{embed})	Embedding space (K, D_{embed})	$\frac{Bits}{SP\ vector}$
MLP-VQ-VAE	(1, 1025)	(1, 615)	(5, 123)	(512, 123)	45
		(1, 625)	(25, 25)	(512, 25)	225
		(1, 615)	(41, 15)	(512, 15)	369
		(1, 615)	(205, 3)	(512, 3)	1845

Figure 4.5 shows the quantization process of the MLP-VQ-VAE. The SP vector is the input of the encoder network to produce the z-latent vector. The reshaping method is applied to the z-latent vector for the reshaped z-latent vectors with the vector length corresponding to the vector length of the designed embedding space. The VQ is applied to transform the continuously reshaped z-latent vectors into a discrete presentation, and the inverse VQ is transformed back to the quantized continuous z-latent vectors as quantized reshaped z-latent vectors. The input of the decoder network is the quantized z-latent vectors that reshape from the quantized reshaped z-latent vectors and reproduce the quantized SP vector.

Figure 4.6 shows the proposed MLP-VQ-VAE training process. In the experiment, the training of MLP-VQ-VAE models used the value of 0.0001 as the learning rate. Adam optimizer was applied to optimize network parameters. The number of training epochs was set to 500,000, and each epoch used the random of eight SP vectors from the SP vector database as a mini-batch. In the training process, the encoder network received the input SP vector and produced the z-latent. The VQ process utilized the z-latents to find the minimum (Euclidian distance) based on Equation 4.4. It returned the index of the vector pattern in the embedding space as the discreet representation of the z-latent. In the inverse VQ process, the obtained index was utilized to choose the corresponded vector pattern in the embedding space to represent the quantized z-latent. The decoder network produced the quantized SP vector from the quantized z-latent. In the end, Equation 4.3 calculated the loss, and the Adam optimizer updated the network parameters consisting of the encoder network, decoder network, and embedding space.

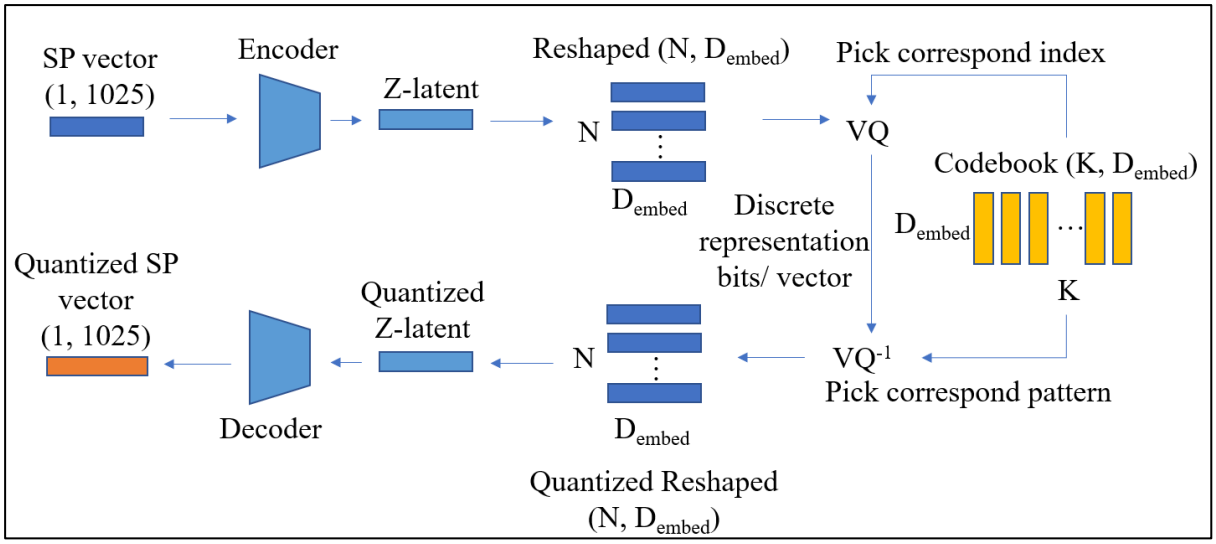

Figure 4.5: The proposed MLP-VQ-VAE diagram.

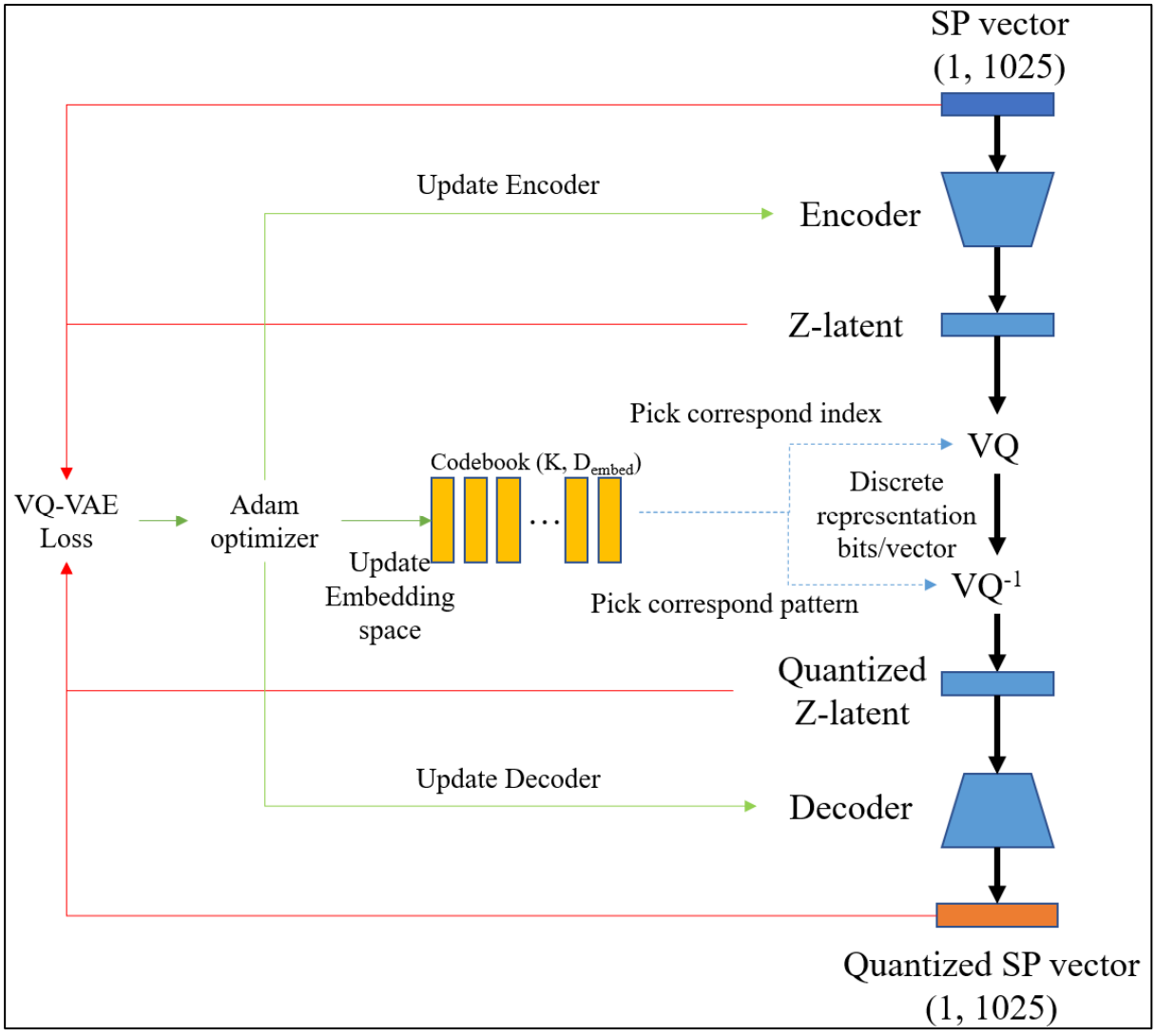

Figure 4.6: The proposed MLP-VQ-VAE training process.

The experimental results calculate the Log Spectral Distortion (LSD) [85, 86] as the spectral envelope distortion indicator; the LSD equation is defined in Equation 4.7:

$$LSD_{(dB)} = 10 \times \frac{2}{M}\sqrt{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left(X_{ij} - Y_{ij}\right)^2}, \tag{4.7}$$

where M is the number of log-spectral coefficients frames, N is the length of each log-spectral coefficient, X_{ij} is the original logarithm with base ten spectral coefficients of the WORLD output waveform, and Y_{ij} is the logarithm with base ten spectral coefficients of the WORLD quantized SP output waveform.

Figures 4.7 and 4.8 show the process of calculating the average LSD for the conventional VQ and the proposed MLP-VQ-VAE, respectively. The testing set consisted of 100 raw speech waveforms from the VCTK corpus that were not included in the training process of the conventional VQ and the proposed MLP-VQ-VAE. The WORLD vocoder extracted the speech parameters from each raw speech waveform, only the spectral envelope parameter (SP) was applied to quantize, and the synthesis process reconstructed the raw speech waveform again based on the quantized SP and other parameters. The LSD was calculated to measure the distortion between SP and quantized SP for each raw speech waveform. Finally, the average LSDs were calculated for each condition.

Figure 4.9 shows the average values of LSD that were evaluated from the four models of both conventional vector quantization and the MLP-VQ-VAE with four target bitrates.

Figure 4.10 shows the examples of the quantized SP vectors for four target bits/SP vector.

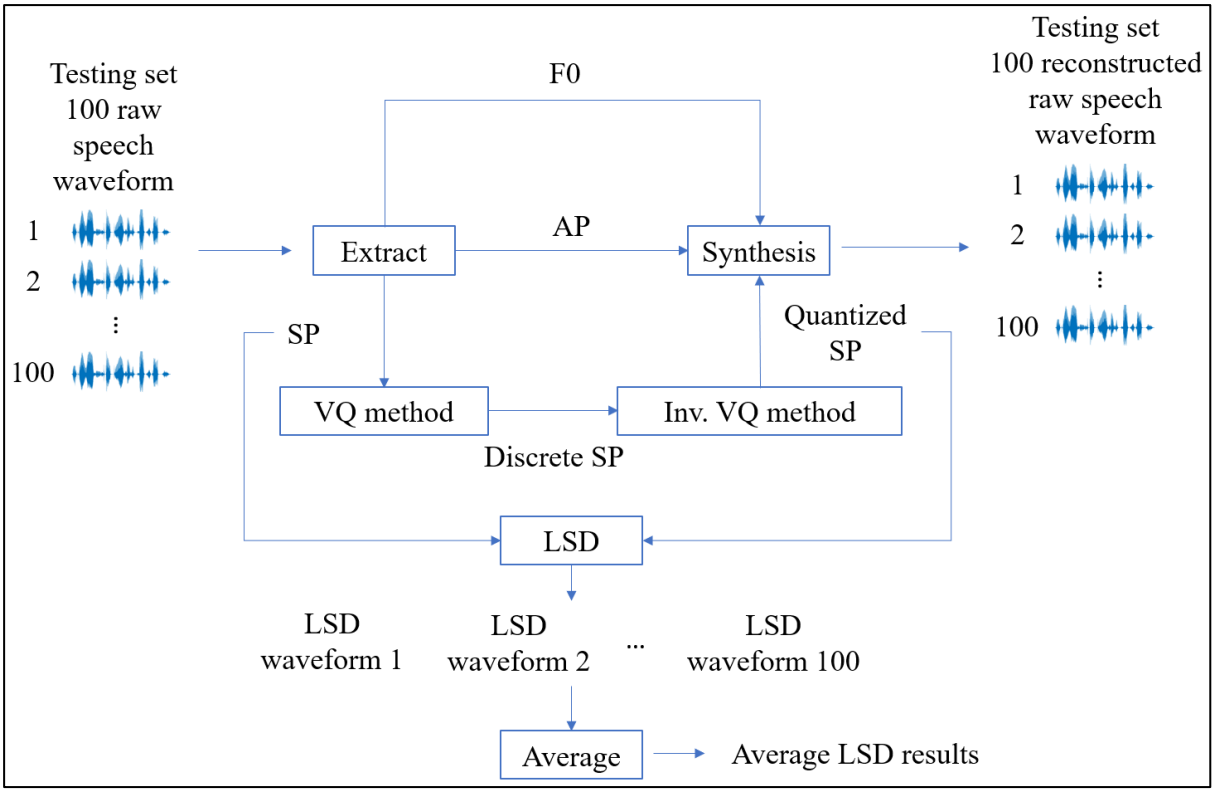

Figure 4.7: The conventional VQ average LSD evaluation.

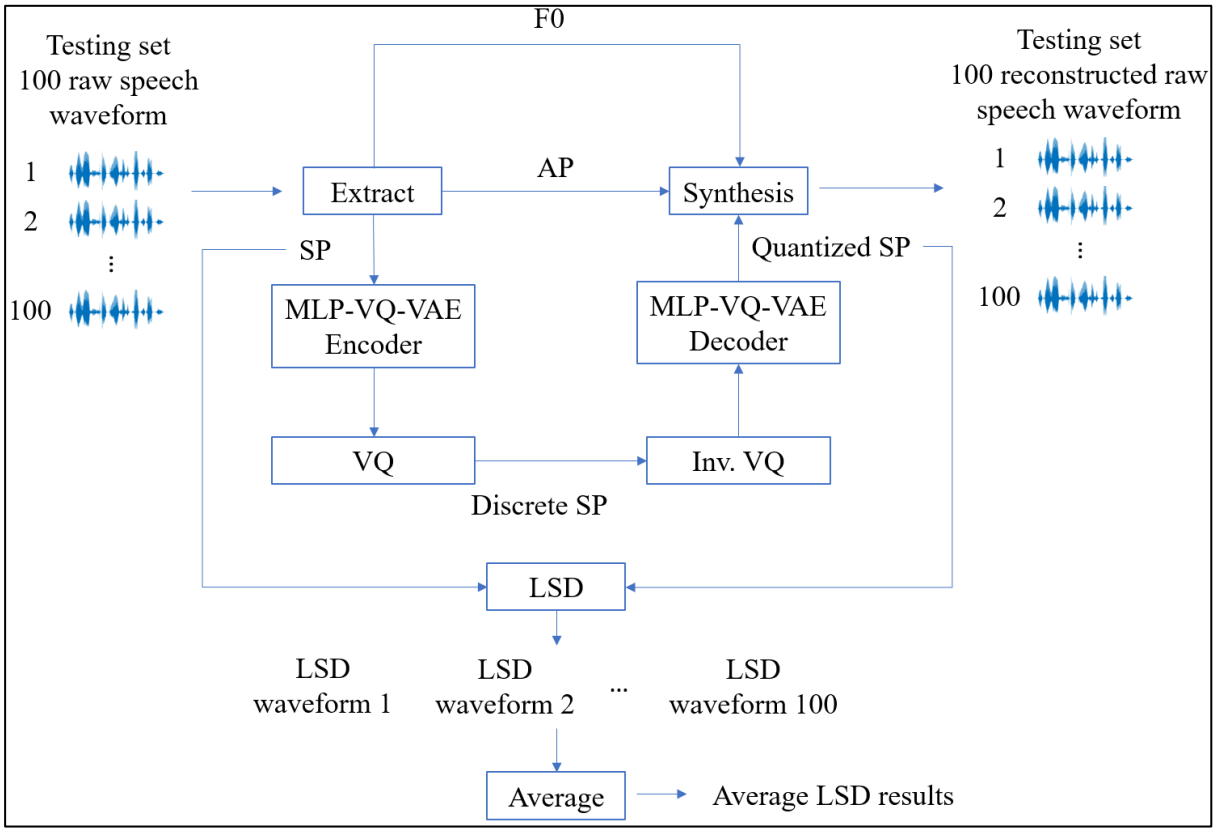

Figure 4.8: The proposed MLP-VQ-VAE average LSD evaluation.

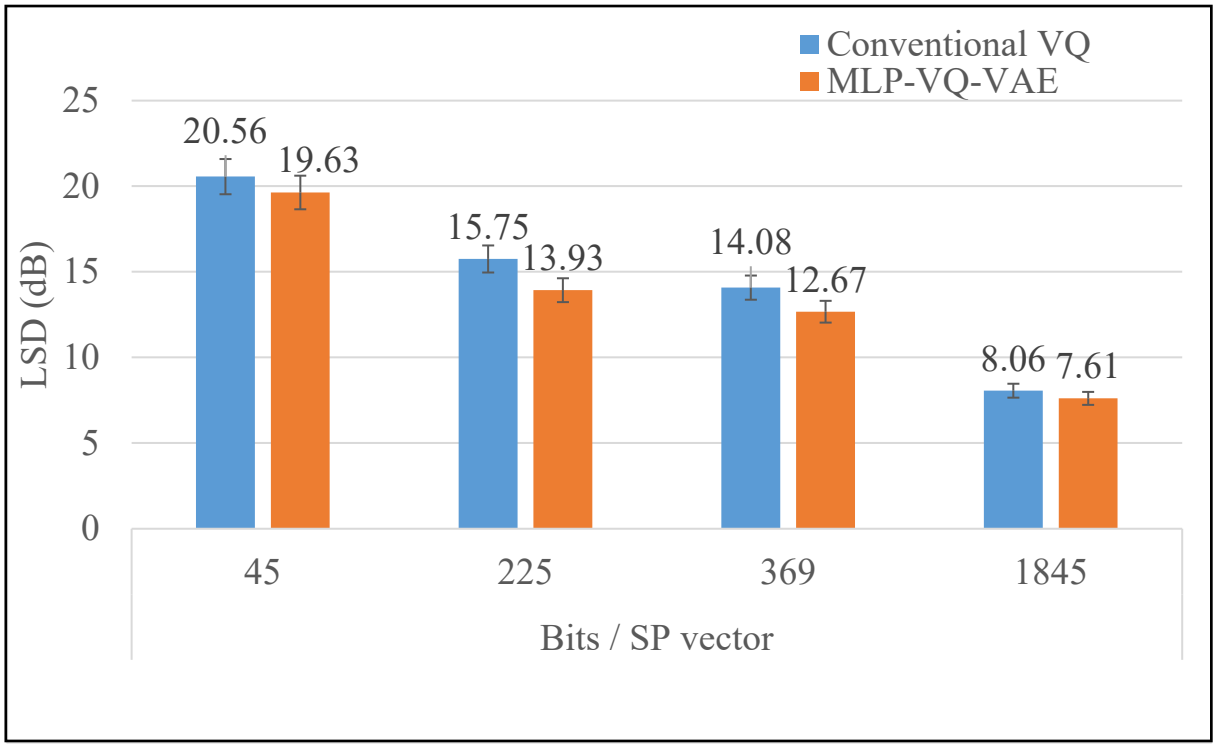

Figure 4.9: The comparison of LSD (dB) in four targets of bits/SP vector, the conventional vector quantization, and the proposed MLP-VQ-VAE.

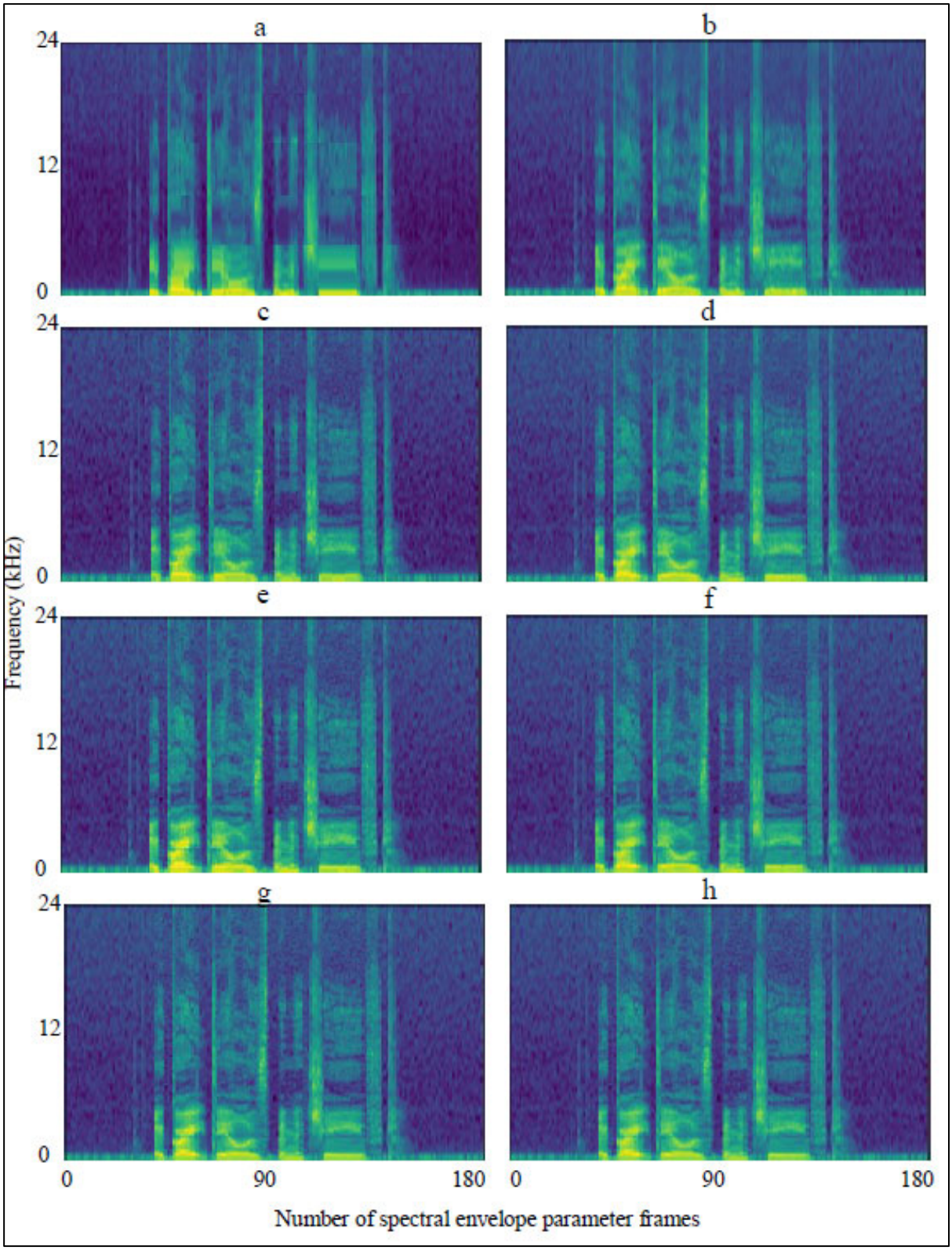

Figure 4.10: The example of quantized spectral envelope parameter frames; (a), (c), (e), (g) are 45, 255, 369, 1845 bits/SP vector of conventional vector quantization, respectively. (b), (d), (f), (h) are 45, 255, 369, 1845 bits/SP vector of MLP-VQ-VAE, respectively.

4.3.4 The comparison of MLP-VQ-VAE and VQ-VAE for spectral envelope quantization

This experiment investigated the performance of quantizing the SP vector of both the MLP-VQ-VAE and the VQ-VAE methods. The embedding space was fixed to the number of K = 512 vector patterns for both techniques. As for the VQ-VAE architecture, the encoder network was implemented with two convolutional layers as (stride 2, 4×4 filter with 64 filter depth, ReLU) and (stride 2, 4×4 filter with 128 filter depth, ReLU), followed by two residual blocks implemented as (stride 1, 3×3 filter with 64 filter depth, ReLU) and (stride 1, 1×1 filter with 128 filter depth, ReLU), connected with one convolutional layer implemented as (stride 1, 1×1 filter with 64 filter depth). The decoder was a counterpart of the encoder network, one transposed convolutional layer (stride 1, 1× 1 filter with 128 filter depth, ReLU) renewed the suitable shape for the same two residual blocks as the encoder network, and followed by the first transposed convolutional layer (stride 1, 4×4 filter with 64 filter depth, ReLU) followed with the second transposed convolutional layer (stride 1, 4×4 filter with 1 filter depth, Sigmoid).

The four models of VQ-VAE were constructed to operate in four target bitrates as shown in Table 4.3. The SP vector length was 1025. The z-latent conducted from the encoder network had a limited shape for reshaping. However, the four reshaped z-latent shapes were (N=8,D_{embed}=2056), (N=32, D_{embed}=514), (N=64, D_{embed}=257), and (N=257, D_{embed}=64), and the output bitrates were 72, 288, 576, and 2313 bits/SP vector for the first to the fourth models, respectively.

Figure 4.11 shows the quantization process of the VQ-VAE. The SP vector was the input of the encoder network to produce the z-latent matrix. The reshaping method was applied to the z-latent matrix for the reshaped z-latent vectors with the vector length corresponding to the vector length of the designed embedding space. The VQ was applied to transform the continuous reshaped z-latent vectors into a discrete presentation, and the inverse VQ was transformed back to the quantized z-latent matrix as quantized reshaped z-latent vectors. The input of the decoder network was the quantized z-latent matrix that was reshaped from the quantized reshaped z-latent vectors and reproduced the quantized SP vector.

Table 4.3
The implementation of four VQ-VAE models

Vector Quantization techniques	SP vector	z-latent	Reshaped z-latent (N, D_{embed})	Embedding space (K, D_{embed})	$\frac{Bits}{SP\ vector}$
VQ-VAE	(1, 1025)	(257, 64)	(8, 2056)	(512, 2056)	72
			(32, 514)	(512, 514)	288
			(64, 257)	(512, 257)	576
			(257, 64)	(512, 64)	2313

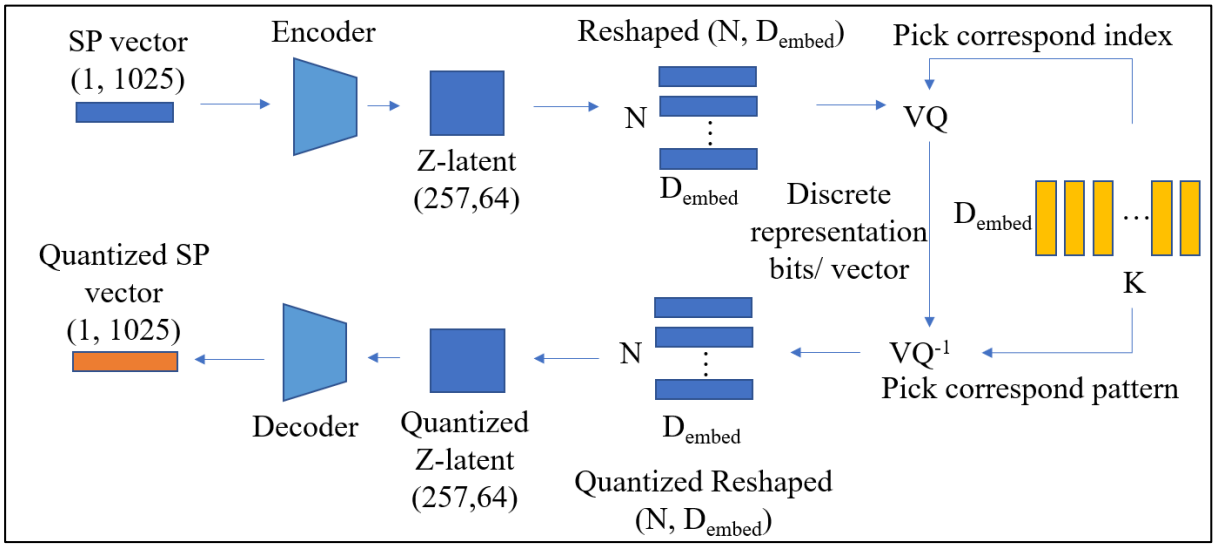

Figure 4.11: The VQ-VAE diagram.

Figure 4.12 shows the VQ-VAE training process. In training, the encoder network receives the input SP vector and produces the z-latent, which minimizes Equation 4.4, and the index is sent to the decoder. The decoder network receives the index and reconstructs the quantized z-latents by picking the corresponding vector pattern in the embedding space. In the end, Equation 4.3 calculates the VQ-VAE loss, and the Adam optimizer updates network parameters consisting of the encoder network, decoder network, and embedding space (codebook). The training of VQ-VAE models used the same parameter values as the MLP-VQ-VAE training. (The learning rate: 0.0001, The optimizer: Adam, The number of training epochs: 500,000, Minibatch: the random of eight SP vectors from the SP vector database.)

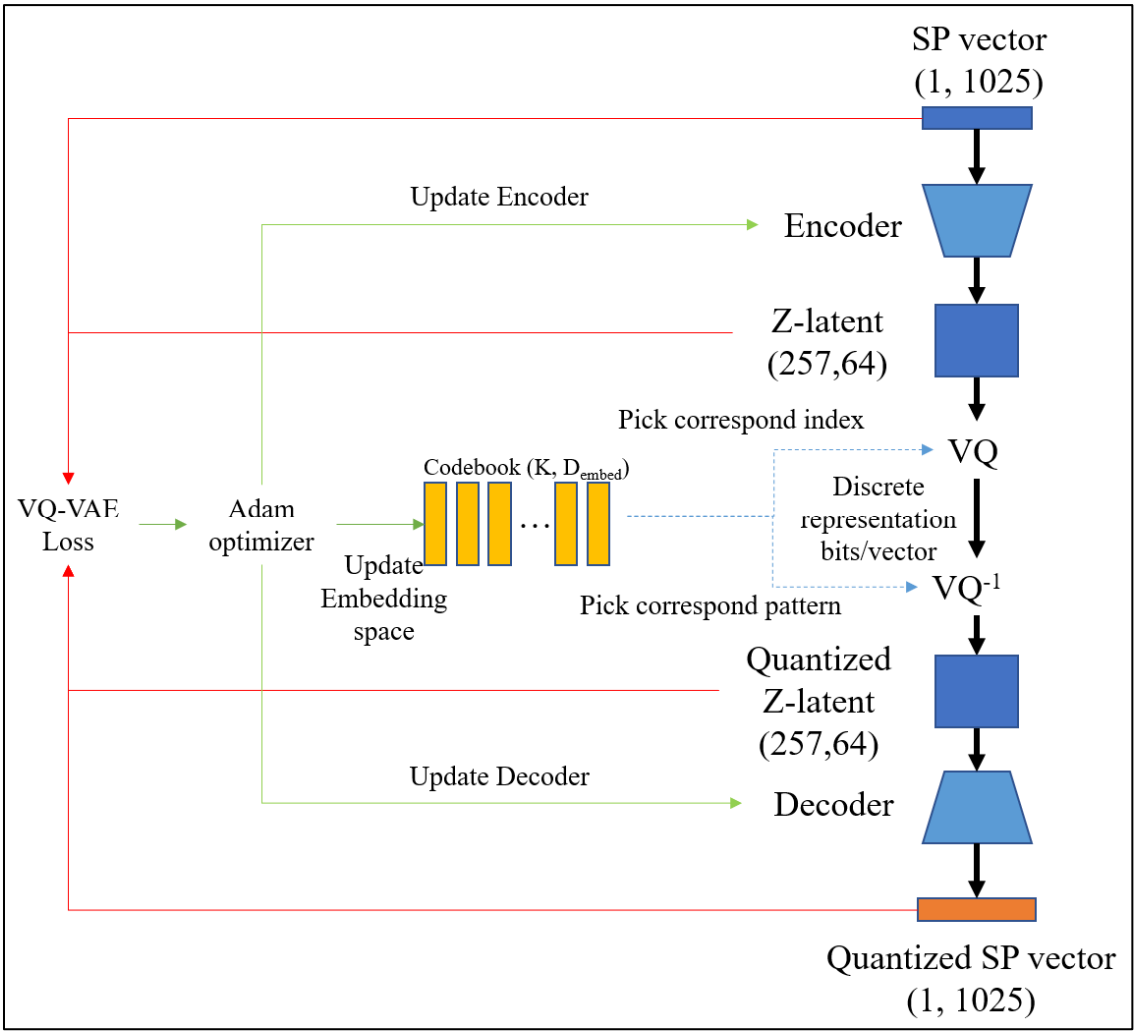

Figure 4.12: The VQ-VAE training process.

The MLP-VQ-VAE architecture compared the previous model with the conventional vector quantization. The coding layer was changed because of the redesign of the expected four target bitrates being the same as the VQ-VAE model. Table 4.4 shows the four models of MLP-VQ-VAE, created with the four target bitrates, corresponding to the VQ-VAE. The encoder networks were 1025-900-768, 1025-900-768, 1025-900- 768, and 1025-900-771, the z-latent vector sizes were (1, 768), (1, 768), (1, 768), (1, 771), the reshaped z-latents are (N =8, D_{embed} = 96), (N = 32, D_{embed}= 24), (N =64, D_{embed}= 12), and (N = 257, D_{embed} = 3), the bitrates became 72, 288, 576, and 2313 bits/SP vector for the first to the fourth models, respectively.

Table 4.4
The implementation of four MLP-VQ-VAE models for compression with VQ-VAE

Vector Quantization techniques	SP vector	z-latent vector	Reshaped z-latent (N, D_{embed})	Embedding space (K, D_{embed})	$\frac{Bits}{SP\ vector}$
MLP-VQ-VAE	(1, 1025)	(1, 768)	(8, 96)	(512, 96)	72
		(1, 768)	(32, 24)	(512, 24)	288
		(1, 768)	(64, 12)	(512, 12)	576
		(1, 771)	(257, 3)	(512, 3)	2313

Figure 4.13 presents the quantization process of the MLP-VQ-VAE. The process was the same as Figure 4.5 in subsection 4.3.3.

Figure 4.14 shows the proposed MLP-VQ-VAE training process. The process was the same as Figure 4.6 in subsection 4.3.3. The MLP-VQ-VAE training parameters were also the same in subsection 4.3.3. (The learning rate: 0.0001, The optimizer: Adam, The number of training epochs: 500,000, Minibatch: the random of eight SP vectors from the SP vector database.)

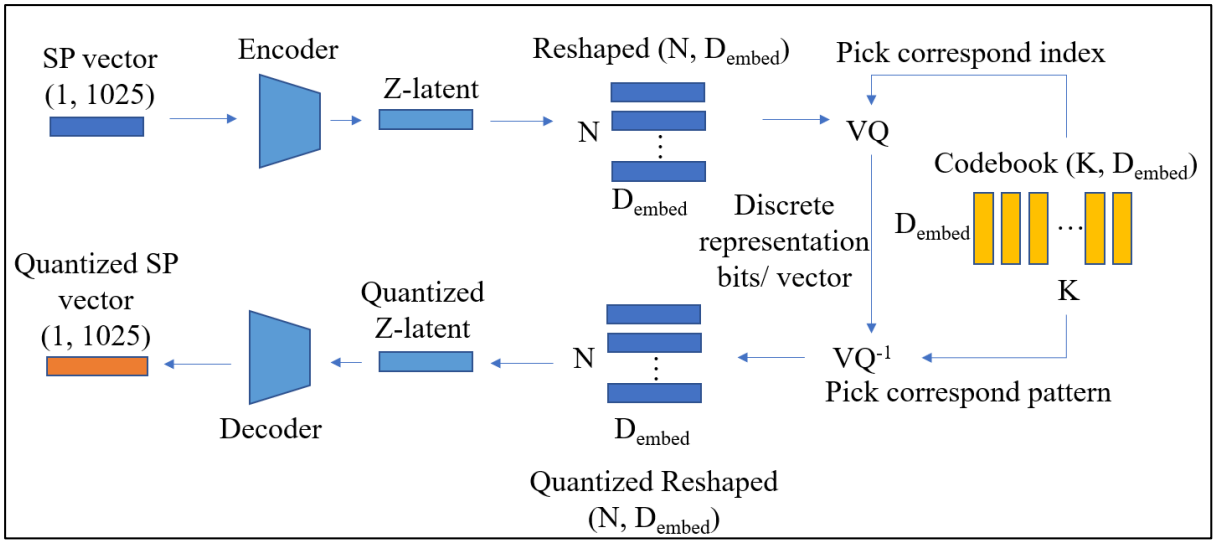

Figure 4.13: The proposed MLP-VQ-VAE diagram.

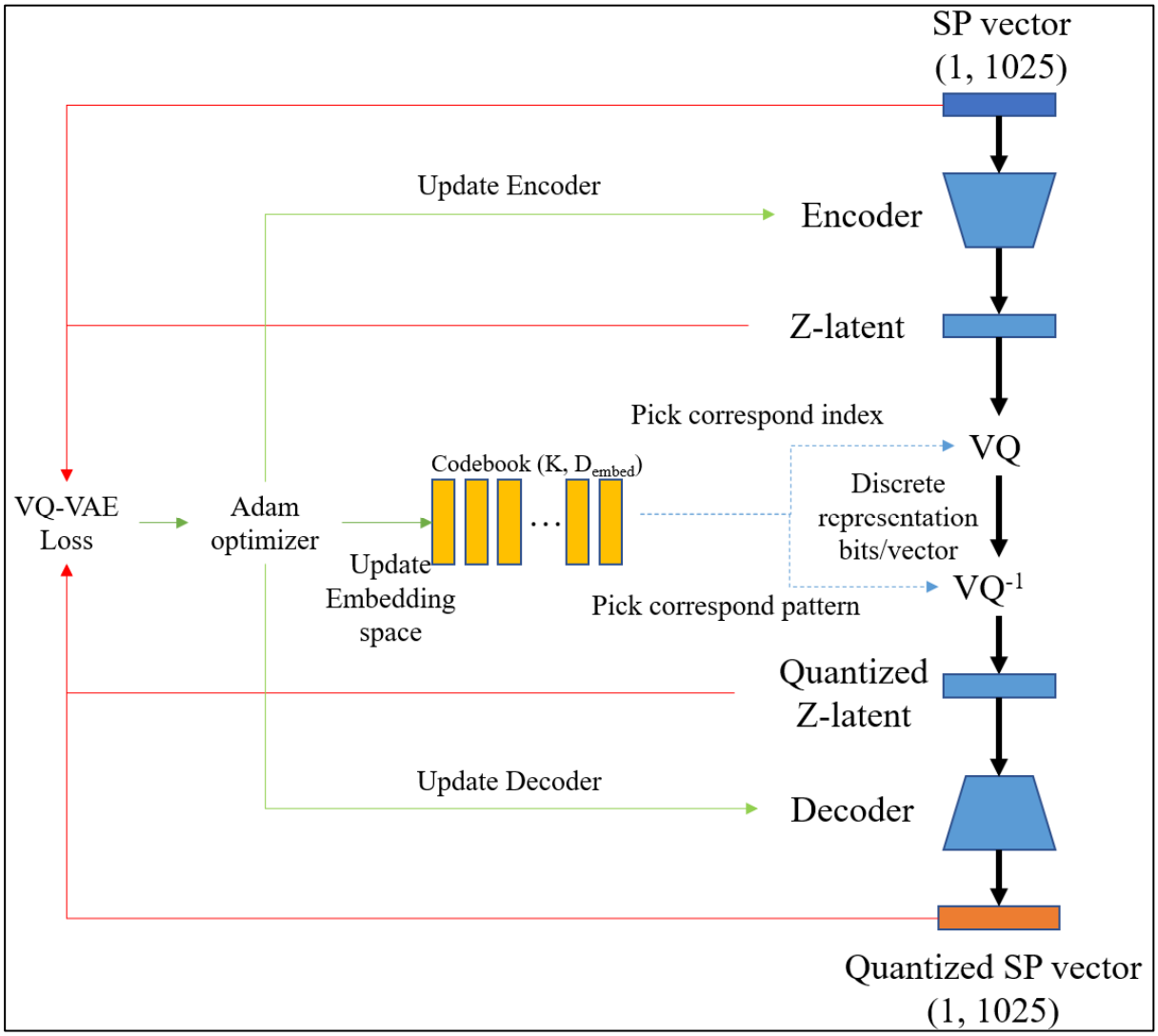

Figure 4.14: The proposed MLP-VQ-VAE training process.

The experimental results for comparing VQ-VAE and the proposed MLP-VQ-VAE calculated the LSD defined in Equation 4.7. Figures 4.15 and 4.16 presented the process of calculating the average LSD for the VQ-VAE and the proposed MLP-VQ-VAE, respectively. These processes were the same, except that the encoder and decoder were either VQ-VAE or MLP-VQ-VAE.

The testing set consisted of 100 raw speech waveforms from the VCTK corpus that were not included in the training process of the VQ-VAE and the proposed MLP-VQ-VAE. The WORLD vocoder extracted the speech parameters from each raw speech waveform, only the spectral envelope parameter (SP) was applied to quantize, and the synthesis process reconstructed the raw speech waveform again based on the quantized SP and other parameters. The LSD was applied to measure the distortion between SP and quantized SP for each raw speech waveform. Finally, the average LSDs of 100 samples were calculated for each condition.

Figure 4.17 shows the LSDs of four models of both VQ-VAE and the MLP-VQ-VAE with four target bitrates. Moreover, Figure 4.18 shows the examples of quantized SP vectors for four target bits/SP vector.

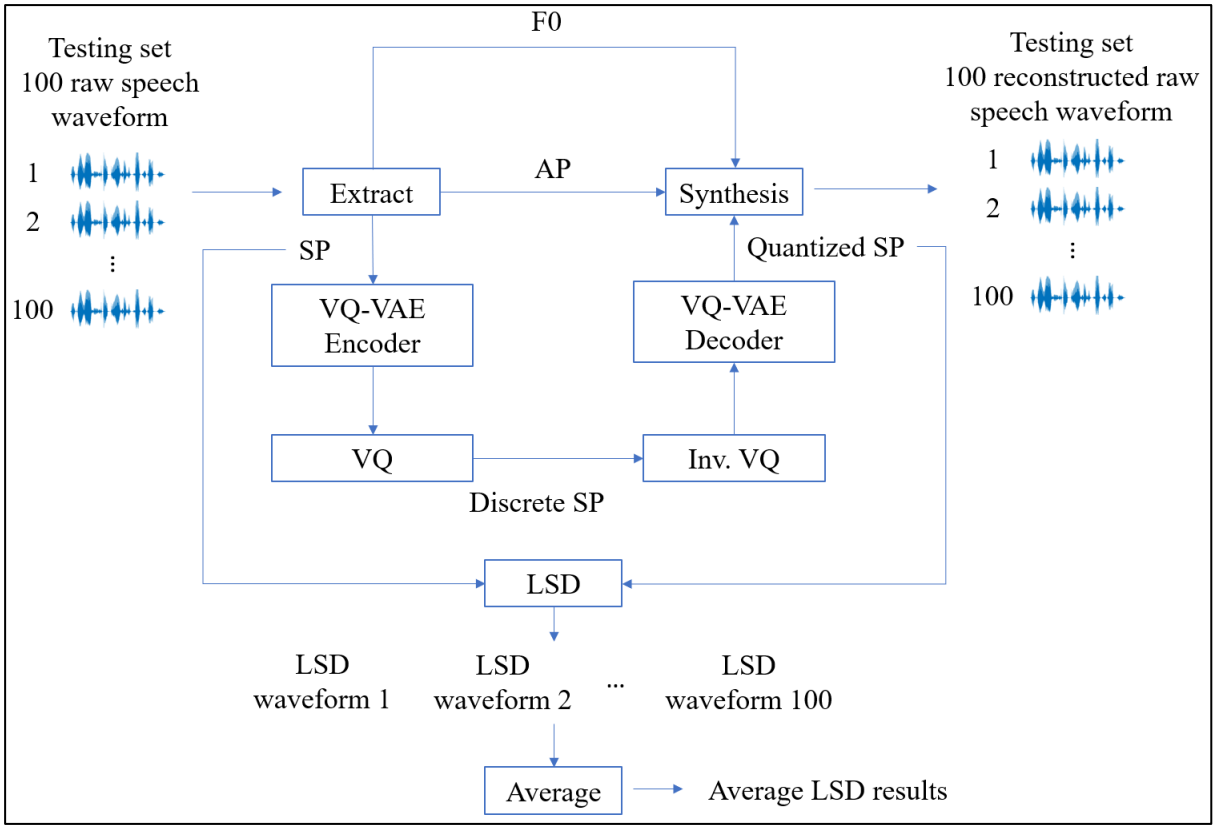

Figure 4.15: The VQ-VAE average LSD evaluation.

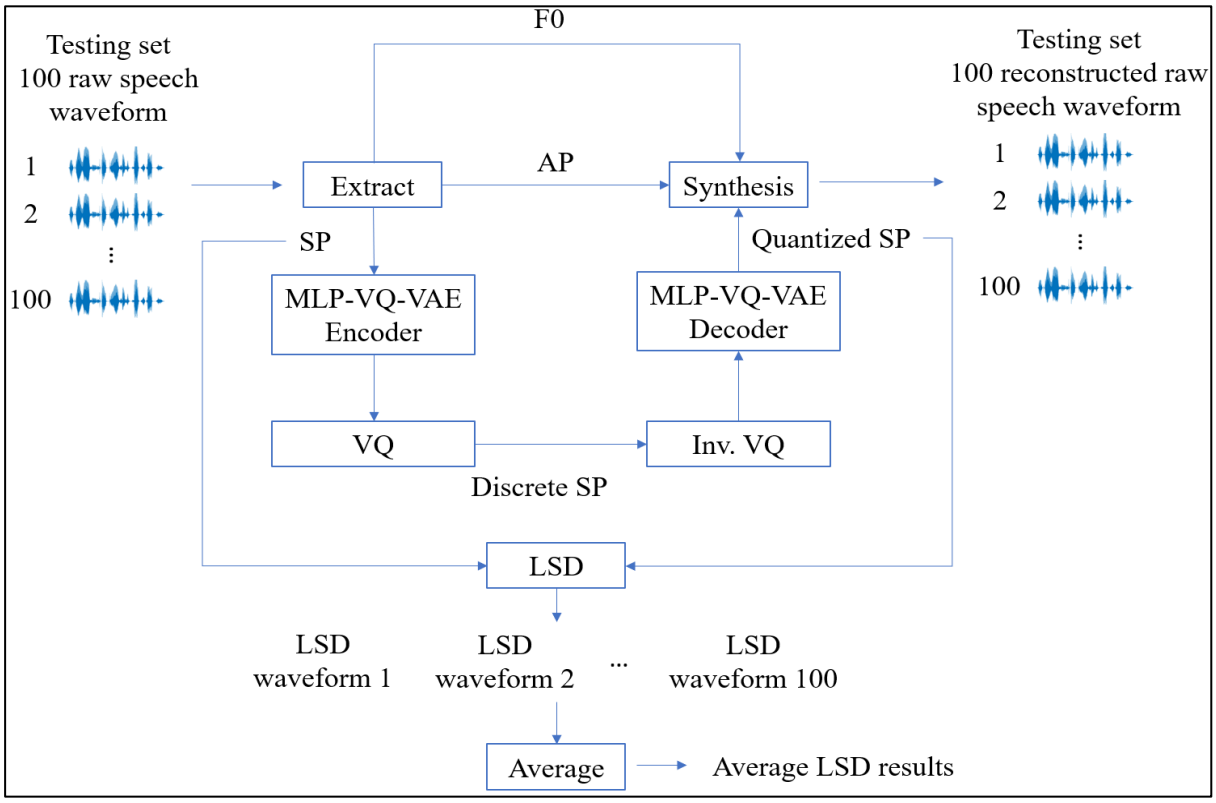

Figure 4.16: The proposed MLP-VQ-VAE average LSD evaluation.

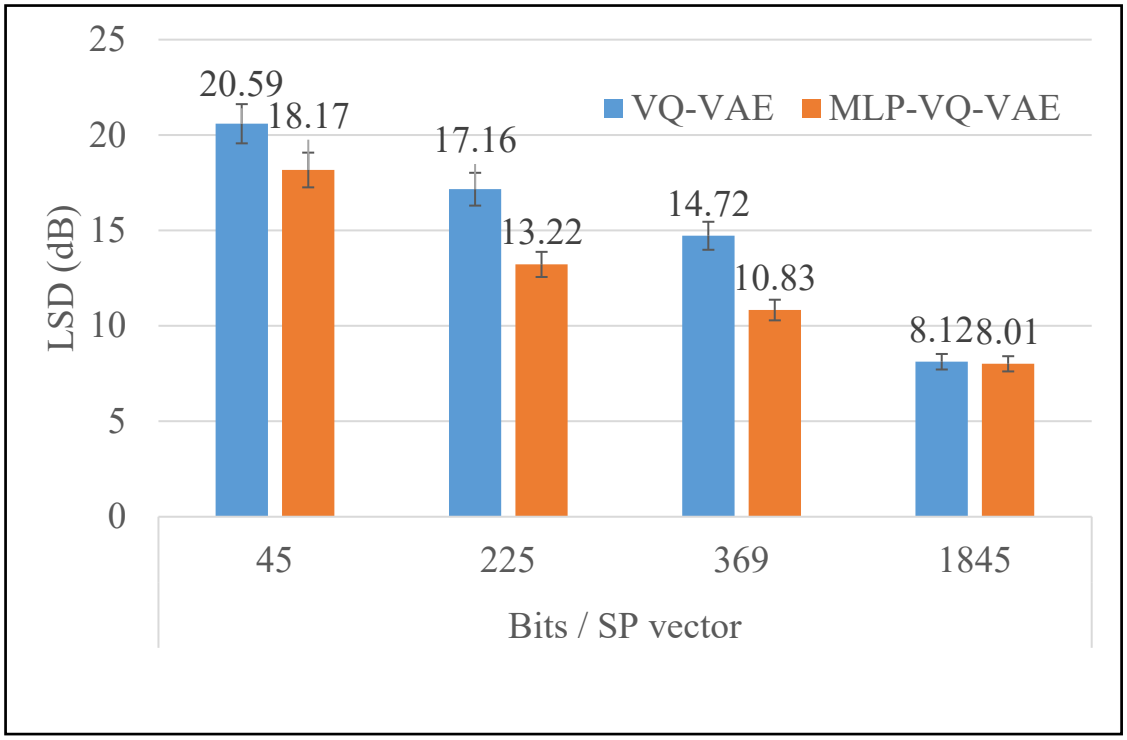

Figure 4.17: The comparison of LSD (dB) in four targets of bits/SP vector, the VQ-VAE, and the proposed MLP-VQ-VAE.

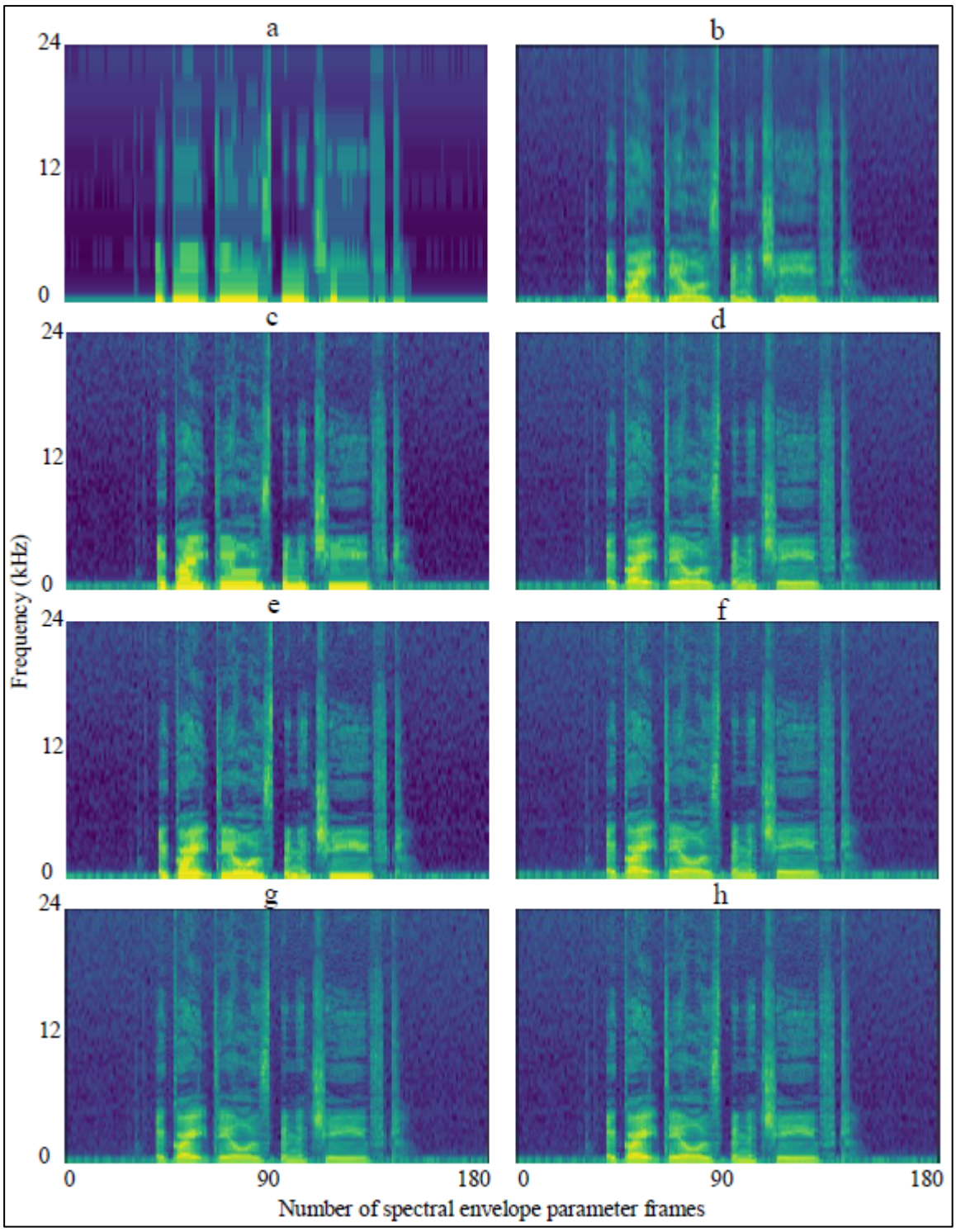

Figure 4.18: The example of quantized spectral envelope parameter frames. (a), (c), (e), (g) are 72, 288, 576, 2313 bits/SP vector of VQ-VAE, and (b), (d), (f), (h) are 72, 288, 576, 2313 bits/SP vector of MLP-VQ-VAE, respectively.

4.4 Discussion

We examined the performance of MLP-VQ-VAE with the flexibility of controlling the z-latent size and the ability of dimensionality reduction. The experiments were designed to quantize the spectral envelope parameters of the WORLD vocoder.

In the first experiment, the conventional vector quantization and the MLP-VQ-VAE were compared for the four target bitrates. The results from Table 4.1 and Table 4.2 show that the MLP-VQ-VAE can decrease the input vector size and the codebook size space by around 1.6 times. From the results in Figure 4.9, the average of the MLP-VQ-VAE LSD gives lower LSDs than those of conventional vector quantization. The average difference for four target bitrates is around 1.1 points in dB.

The second experiment was the comparison of VQ-VAE and the MLP-VQ-VAE. The results from Table 4.3 and Table 4.4 express that the MLP-VQ-VAE can decrease the z-latent and the embedding space size by around 21.4 times. The MLP-VQ-VAE also points out the significant ability to reduce LSD distortion. The average from four target bitrates is around 2.5 points in dB, from the results shown in Figure 4.17.

The MLP-VQ-VAE has shown an effective way to design the z-latent in a flexible size, corresponding to the embedding space size. As results have shown, the smaller reshaped z-latent vector takes effect to give lower LSD than the longer reshaped z-latent vector in the vector quantization process. As shown in Figure 4.10 and Figure 4.17, the MLP-VQ-VAE has a better ability to recover the spectral envelope parameters, especially at lower bitrates. The MLP-VQ-VAE has less computational complexity than the VQ-VAE and is also flexible in designing the end-to-end quantization system based on deep learning in the embedded system.

4.5 Conclusion

In conclusion, this chapter provides the following contributions:

- The MLP-VQ-VAE has been proposed to quantize the spectral envelope parameters of the high-quality 48kHz WORLD vocoder.
- The MLP-VQ-VAE has a smaller codebook size, around 1.6 times associated with a conventional vector quantization method and around

21 times smaller in the embedding space and z-latent sizes compared with the VQ-VAE.

- In terms of the Log Spectral Distortion (LSD), the average results from four operation bitrates, the MLP-VQ-VAE has reduced by around 1.1 points in dB compared to conventional vector quantization and around 2.5 points in dB compared to the VQ-VAE.

Chapter 5

The effect of vector quantization techniques in vector quantization based on deep learning for speech spectral envelope quantization

5.1 Overview

The objective of this section is to study two techniques utilized in the conventional VQ [40, 41], the Sub-band VQ [76, 77], and the predictive VQ [79, 80] for VQ-VAE deep-learning based architectures. When applying those techniques in VQ-based learning, we investigate the reconstruction performance.

Figure 5.1 shows an overview of the spectral envelope quantization of the conventional vector quantization techniques in Vector Quantized Variational AutoEncoder (VQ-VAE) [44, 45]. We propose the Sub-band VQ-VAE, a sub-band quantization technique [76, 77] applied to the VQ-VAE. This model can concentrate on the specific frequency bands to assign more bits and leave the unnecessary band with fewer bits. The predictive vector quantization technique is also investigated in the VQ-VAE and presented as the Predictive VQ. We propose the Predictive VQ-VAE and examine whether the method can efficiently predict the current data from the previous data.

We conducted the quantization experiments for the WORLD [75] spectral envelope parameters from 48 kHz raw speech data. The Sub-band VQ-VAE and the Predictive VQ-VAE were implemented in four target bitrates to quantize the spectral envelope parameters and compared with the same four target bitrates of the plain VQ-VAE.

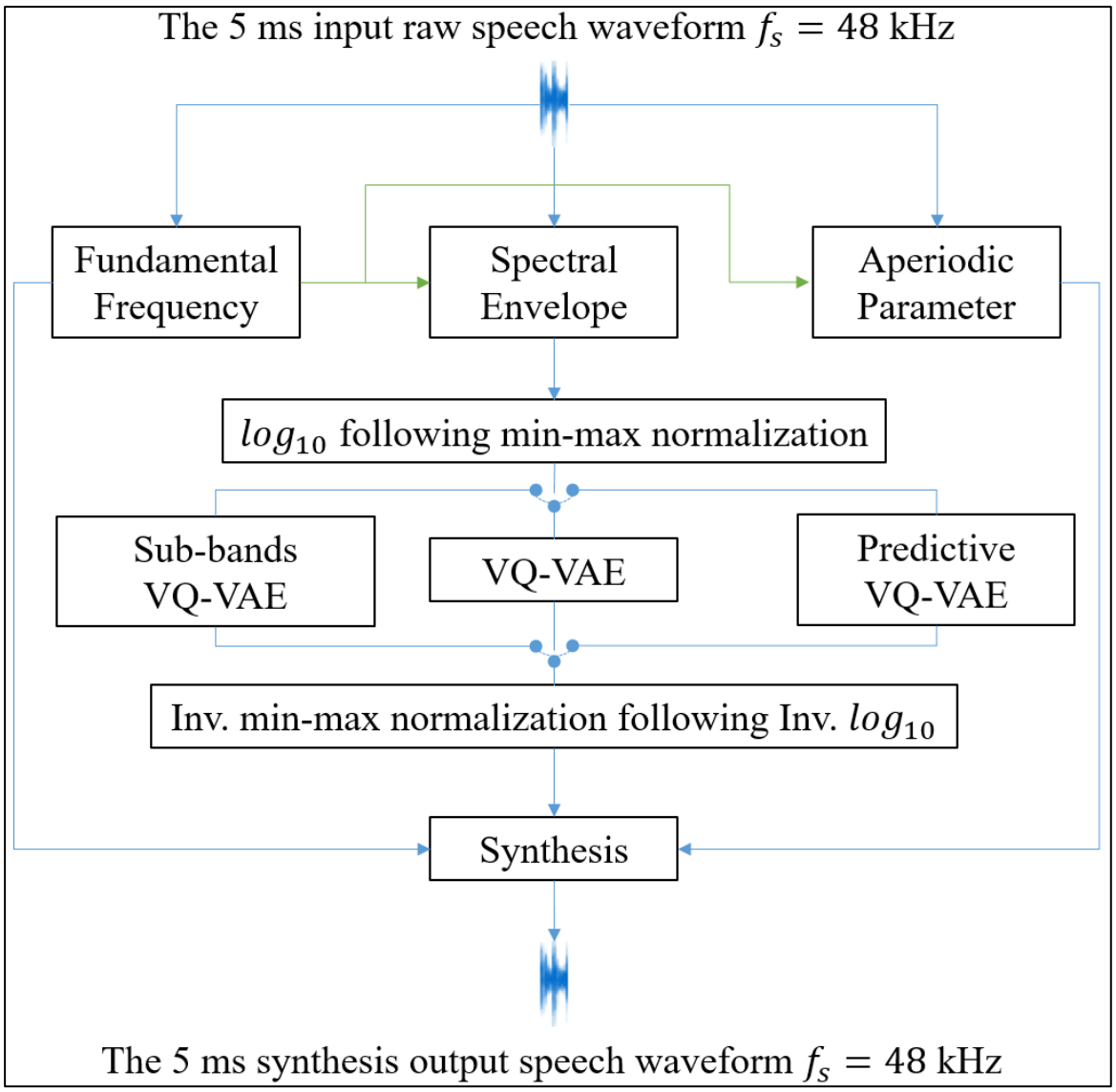

Figure 5.1: Overview of the proposed Sub-band VQ-VAE and the Predictive VQ-VAE compared to the VQ-VAE for the WORLD vocoder spectral envelope quantization.

5.2 Methodology

5.2.1 Vector Quantized Variational AutoEncoder (VQ-VAE)

The Vector Quantized Variational AutoEncoder (VQ-VAE) [44, 45] has been proposed as a modification of the VAE [19]. VQ-VAE substitutes continuous z-latent variables with discrete ones. VQ techniques inspire the model system to cooperate with the autoencoder. The VQ-VAE is unsupervised learning constructed with the simple idea of convolutional neural networks (CNN).

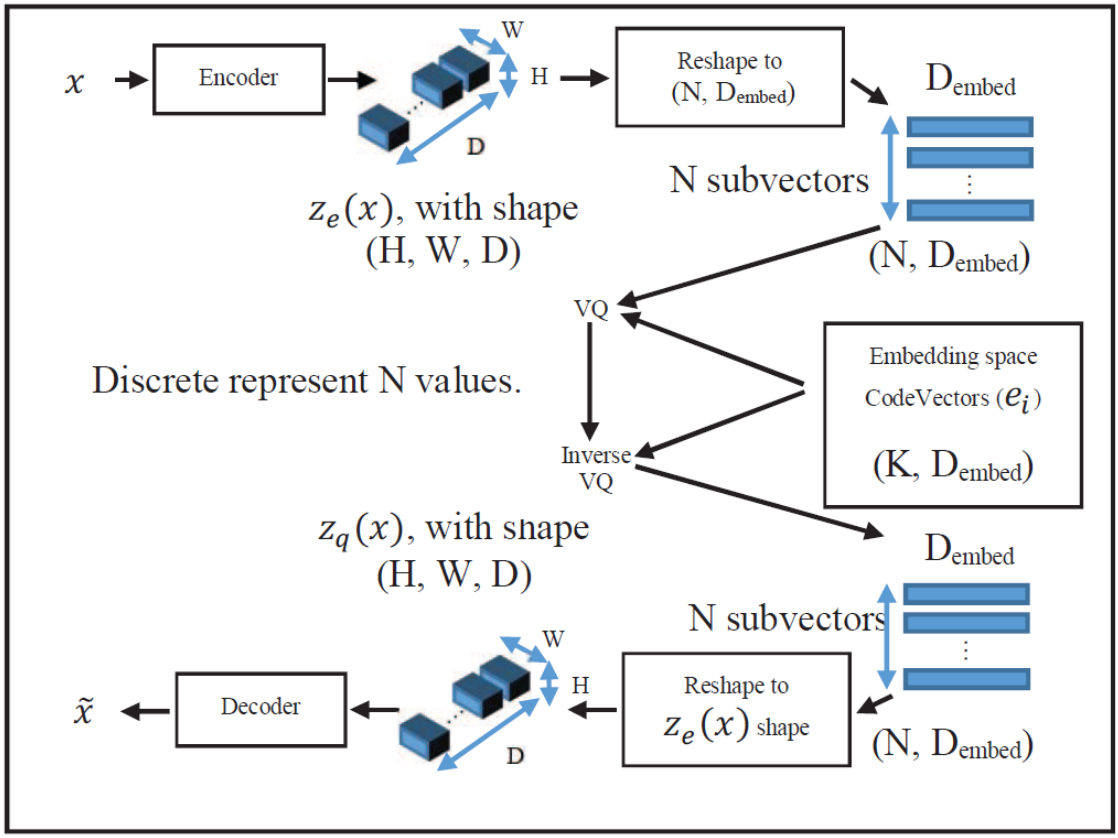

Figure 5.2: The VQ-VAE model.

Figure 5.2 shows the VQ-VAE model. The three layers of CNN construct the encoder networks with stride two and two blocks of CNN residual techniques to produce continuous z-latent $z_e(x)$ variables from input x with shape (H, W, D), where H is the height, W is the width, and D is the depth dimensions. The embedding space keeps the number of prototype vectors $\{ e_i, i = 1, \ldots, K\}$. During the VQ forward pass, the $z_e(x)$ should be reshaped into sub-N vectors of length D_{embed}. At this point, we must assign the finite number of N vectors because the vector must have the same length of D_{embed}. The reshaped z-latent is constructed here by using the VQ method. Each vector in reshaped z-latent is replaced with its closest vectors in embedding space by $q(x) = \min (z_e(x) - e_i)$ to represent the discrete z-latent variable with a set of indices $q(x)$. The inverse quantization generates a reshaped z-latent = $e_{q(x)}$ with the shape of (H, W, D) to represent the quantized z-latent $z_q(x)$.

The decoder networks are the counterpart of encoder networks created by two blocks of CNN residual technique followed by three layers of transposed CNN stride two. The output $\tilde{x}$ of the decoder networks indicates that the input can be reconstructed by using the $z_q(x)$ passed to the decoder

networks as input. During the backward pass, the gradient of the loss concerning updating model parameters is shown in Equation 5.1, and the overall loss function is the sum of three-loss terms. The first term is the negative log-likelihood of the reconstruction, used to carry the gradient from the decoder networks to the encoder networks. The second and the last terms are Euclidean distance loss and the commitment loss, respectively, to update the prototype vectors in embedding space.

$$L = -\log p\,(x|z_q(x)) + \left\|sg[z_e(x)] - e_{q(x)}\right\|_2^2 + \beta\left\|z_e(x) - sg\left(e_{q(x)}\right)\right\|_2^2, \quad (5.1)$$

where x is input, $z_e(x)$ is the z-latent, e is the embedding space, $q(x)$ is the set of nearest vectors index, $z_q(x)$ is the quantized z-latent, and γ is the commitment cost set to 0.25.

5.2.2 The Sub-band Vector Quantized Variational AutoEncoder (Sub-band VQ-VAE)

Figure 5.3 presents the Sub-band Vector Quantized Variational AutoEncoder (Sub-band VQ-VAE) [76, 77] that has been inspired by sub-band coding techniques. From the Nyquist theorem, the sampling frequency must be two times the information frequency. That means the raw speech waveform with a 48 kHz sampling frequency has the maximum speech information frequency at 24 kHz. The WORLD vocoder works with the 48 kHz sampling raw speech waveform.

The SP of WORLD vocoder [75] has extracted every 240 samples from 48 kHz raw speech waveforms. The SP vector has a length of 1025 with speech information from 0 to 24 kHz. In general, most speech information is kept in a frequency below 16 kHz. Then, we divide the SP vector into two sub-vectors. The first vector with the length of 684 is the lower band frequency vector that represents the speech information from 0 to 16 kHz, which corresponds to the first 684 values of the SP vector. The second vector with the length of 341 is the higher band frequency vector representing the speech information from 16 kHz to 24 kHz, which corresponds to the value number 685 to 1025 of the SP vector. As the cost function, the same cost of Equation 5.1 is used to train network parameters.

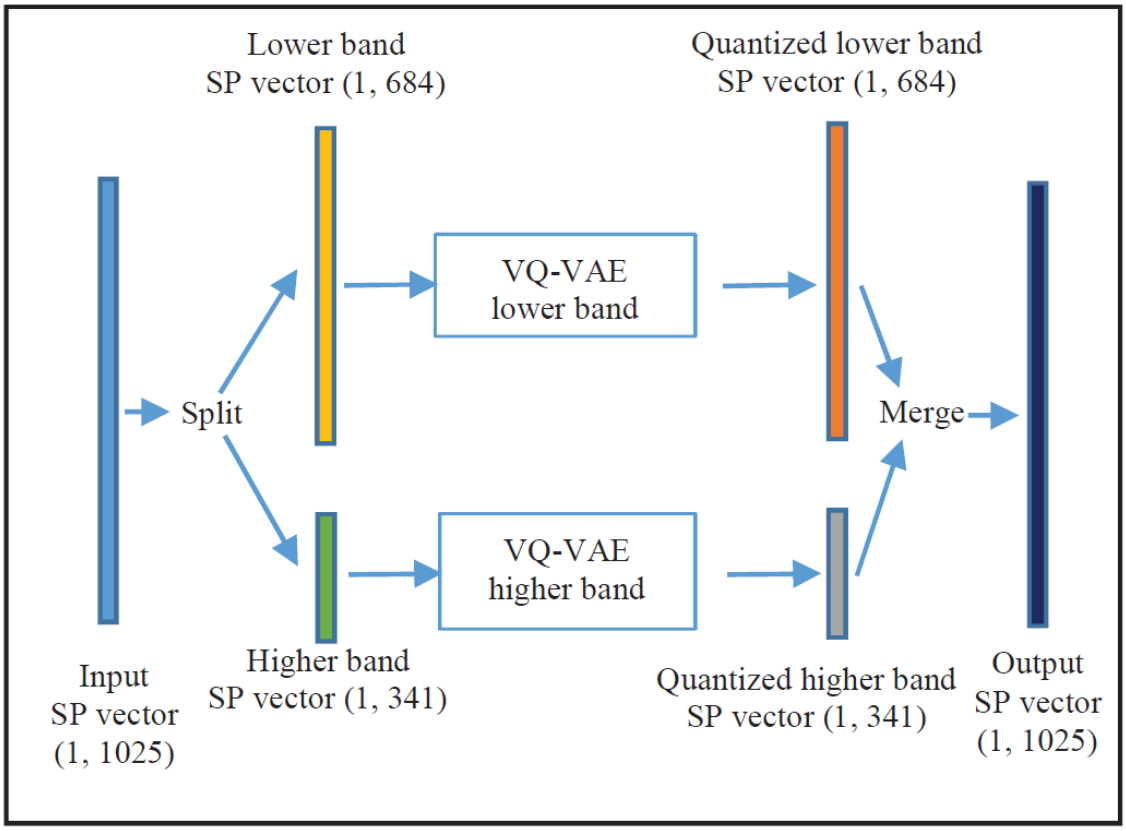

Figure 5.3: The Sub-band Vector Quantized Variational AutoEncoder.

5.2.3 The Predictive Vector Quantization Variational AutoEncoder (Predictive VQ-VAE)

The Predictive Vector Quantized Variational AutoEncoder (Predictive VQ-VAE) [79, 80] introduces the PVQ technique into the VQ-VAE. Figure 5.4 shows the prediction model. In the encoding process, the input data x is fed into the encoder network to produce the z-latent $z_e(x)$. It is reshaped in the N sub-vector (reshaped $z_e(x)$) corresponding to the designed embedding space (K= 2^n bits, D_{embed}). The quantization process creates the discrete representation i and the quantized z-latent $z_q(x)$ is the input to the encoder predictor network. The predicted z-latent $\tilde{z}_q(x)$ is subtracted from the next $z_e(x)$ and added to the next $z_q(x)$. In the decoding process, the received discrete representation i reproduces the quantized z-latent $z_q(x)$. The decoder predictor network employs the $z_q(x)$ as the input to produce the $\tilde{z}_q(x)$ to add to the next $z_q(x)$, and the VQ-VAE decoder network incorporates $z_q(x)$ to reconstruct the output data $\tilde{x}$ corresponding to the input data x. As the cost function, the same cost as Equation 5.1 is applied to train the network parameters.

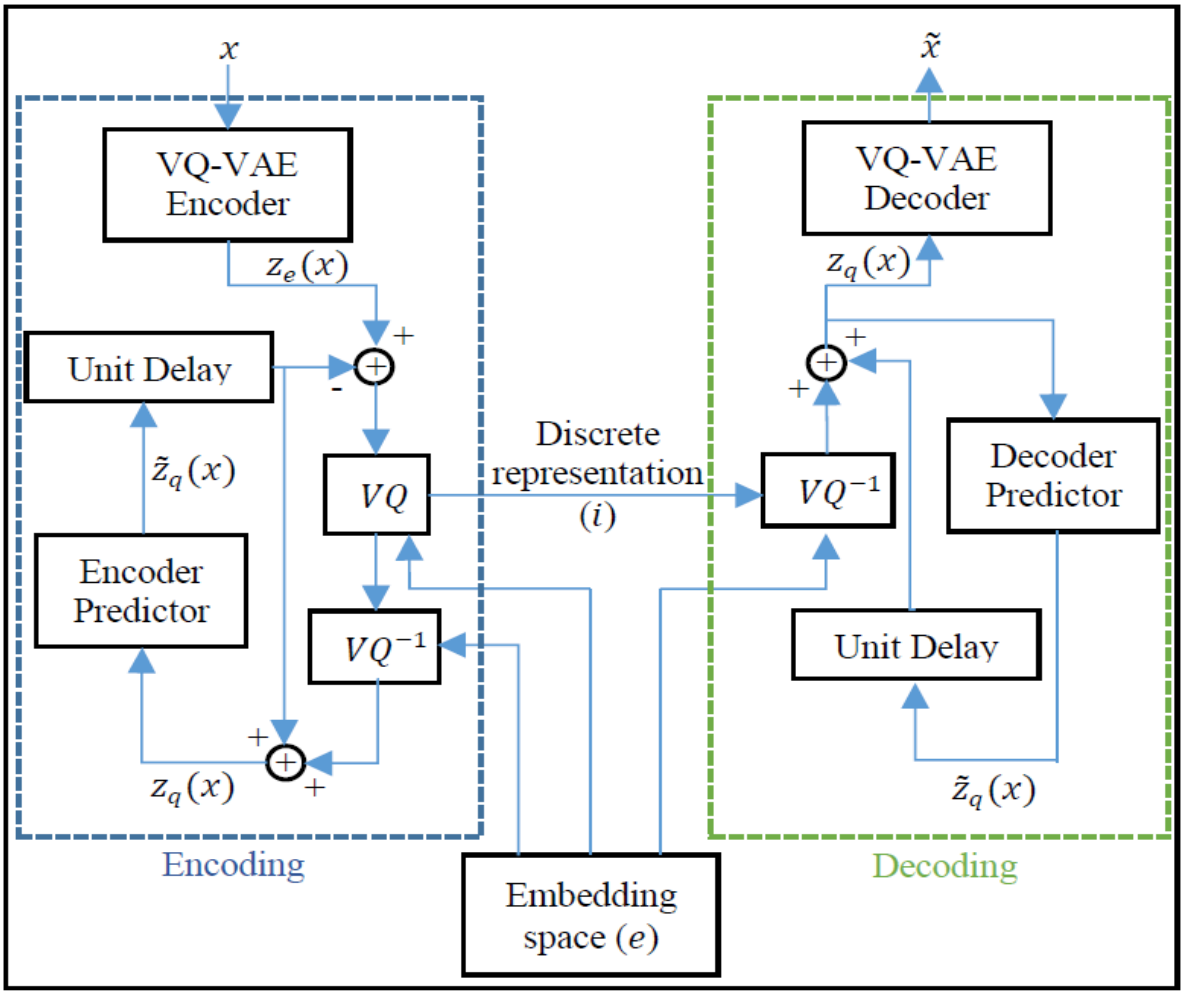

Figure 5.4: The Predictive Vector Quantized Variational AutoEncoder.

5.3 Experiments and Results

5.3.1 WORLD vocoder spectral envelope quantization

In the same manner, as the chapter 4, we investigated the spectral envelope quantization performance obtained by the WORLD vocoder. The sampling frequency of the input raw speech was 48 kHz, and at five milliseconds analysis-window shifts. First, the spectral envelope parameter (SP) with the length of 1025 was calculated at the encoder. Next, the SP vector was applied to the logarithm with base 10, followed by the min-max normalization to scale values between 0 and 1. Then, the processed data were quantized by the Sub-band VQ or the predictive VQ. At the decoder, The quantized SP vector was recovered by the reversed processes, and the WORLD decoder synthesized the five milliseconds output speech waveform frame by frame.

5.3.2 Raw speech waveform database

As for the raw input speech waveform database, CSTR VCTK corpus [90] was used. The data format was 16 bits with 48 kHz as sampling frequency, recorded from 109 English native speakers with about 400 sentences with various accents.

5.3.3 The Sub-band Vector Quantized Variational AutoEncoder for Spectral Envelope Quantization

The Sub-band VQ-VAE is a split version of the VQ-VAE and compared with the reference VQ-VAE. First, we explain the reference VQ-VAE. Figure 5.5 shows the quantization process of the VQ-VAE. The SP vector is the input to the encoder network to produce the z-latent matrix. The z-latent matrix is reshaped as the reshaped z-latent vectors with the vector length corresponding to the vector length of the designed embedding space. The VQ is applied to transform the continuous reshaped z-latent vectors into a discrete presentation, and the inverse VQ transforms back the quantized continuous z-latent matrix as quantized reshaped z-latent vectors. The input to the decoder network is the quantized z-latent matrix reshaped from the quantized reshaped z-latent vectors, and the quantized SP vector produced.

Each the VQ-VAE encodes lower and higher band SP vectors. We apply a standard architecture of the VQ-VAE model for quantizing all sub-band SP vectors shown in Table 5.1. The CNN parameter W is a weighting filter with shape (height, width, input_channel, output_channel), B is Bias with (output_channel), and stride is a step to move the W filter according to the input. The encoder networks consist of convolution of three layers with a rectified linear unit (RELU) activation function. The layers are Convolution Layer 1, Convolution Layer 2, and Convolution Prepare Layer to adapt data for the residual technique. Then, they are connected with two blocks of residual techniques, which implement as two layers of convolution networks with the RELU activation function. The last layer of the encoder is the Convolution Prepare VQ Layer without the activation function to produce the z-latent. The decoder networks are the counterpart of the encoder networks constructed with Convolution Prepare Layer with RELU activation function for two blocks of residual techniques followed with two layers of Transposed Convolution Layer with RELU activation function, where the last Transposed Convolution Layer uses sigmoid as an activation function. The embedding

space prototype vectors are set to 512 vectors (9 bits = the number of prototype vectors).

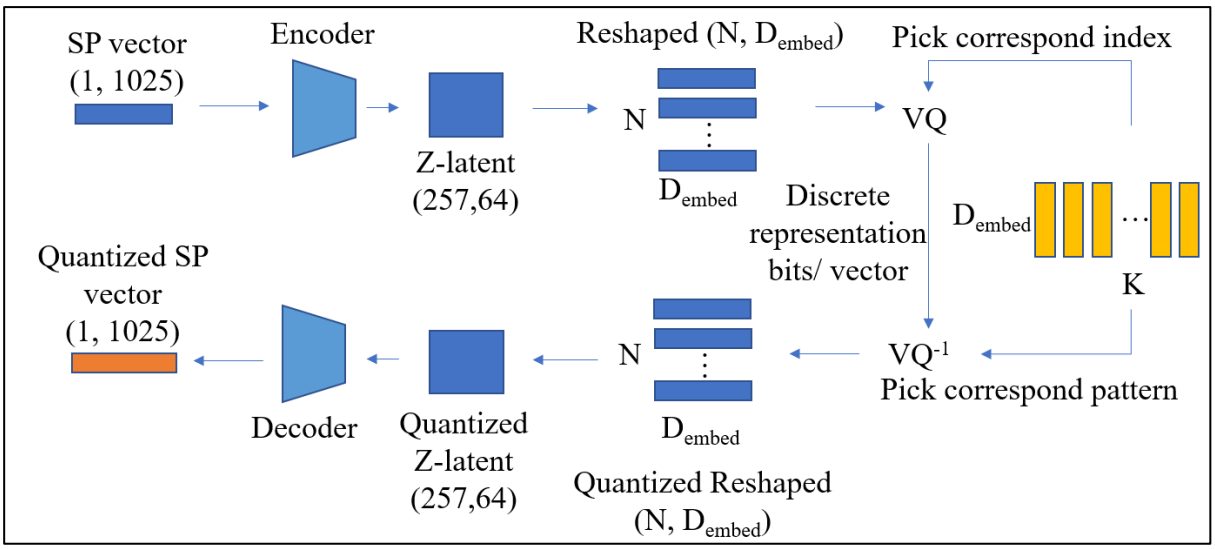

Figure 5.5: The VQ-VAE quantization process.

Table 5.1
The VQ-VAE Architecture used for Sub-band VQ-VAE.

VQ- VAE Architecture		
Encoder	Convolution Layer 1	W(4, 4, 1, 64), B(64), stride(2, 2), RELU
	Convolution Layer 2	W(4, 4, 64, 128), B(128), stride(2, 2), RELU
	Convolution Prepare Layer	W(3, 3, 128, 128), B(128), stride(1,1), RELU
	Convolution Residual 1 Layer 1	W(3, 3, 128, 64), B(64), stride(1,1), RELU
	Convolution Residual 1 Layer 2	W(1, 1, 64, 128), B(128), stride(1,1), RELU
	Convolution Residual 2 Layer 1	W(3, 3, 128, 64), B(64), stride(1,1), RELU
	Convolution Residual 2 Layer 2	W(1,1,64,128), B(128), stride(1,1), RELU
	Convolution Prepare VQ Layer	W(1,1,128,64), B(64), stride(1,1))
Decoder	Convolution Prepare Layer	W(3,3,64,128), B(128), stride(1,1), RELU
	Convolution Residual 1 Layer 1	W(3,3,128,64), B(64), stride(1,1), RELU
	Convolution Residual 1 Layer 2	W(1,1,64,128), B(128), stride(1,1), RELU
	Convolution Residual 2 layer 1	W(3,3,128,64), B(64), stride(1,1), RELU
	Convolution Residual 2 layer 2	W(1,1,64,128), B(128), stride(1,1), RELU
	Convolution Transpose layer 1	W(4,4,128,64), B(64), stride(2,2), RELU
	Convolution Transpose layer 2	W(4,4,64, 1), B(1), stride(2,2), SIGMOID

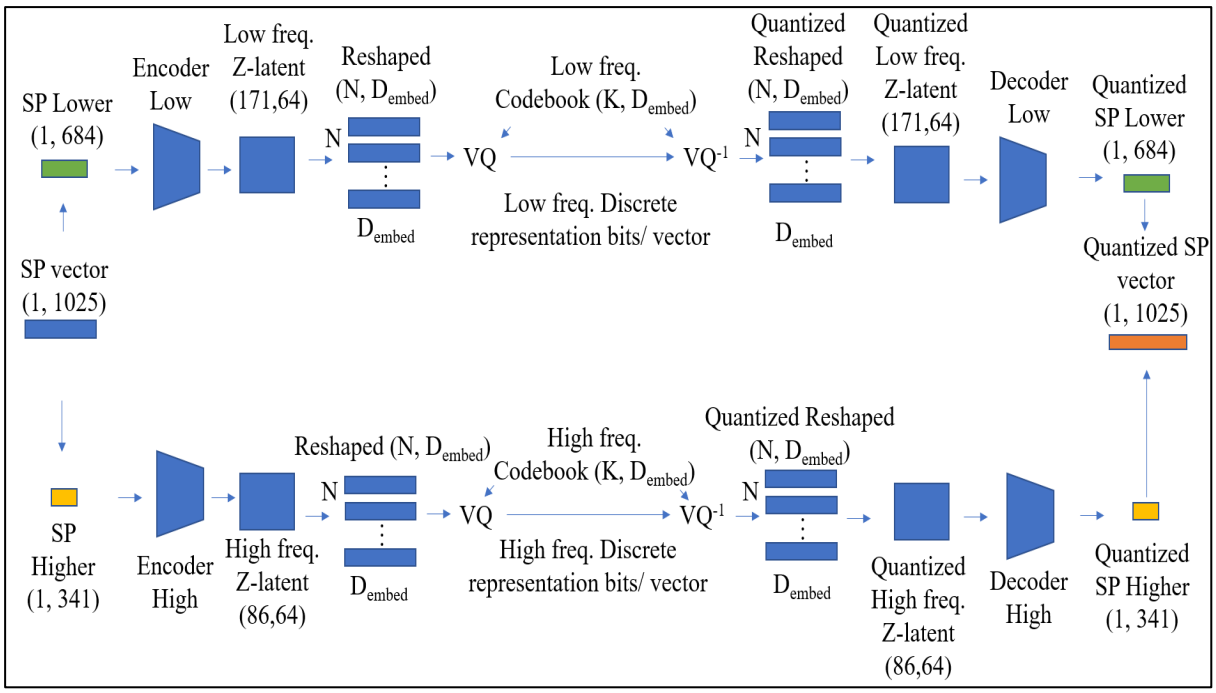

Figure 5.6: The Sub-band VQ-VAE quantization process.

Figure 5.6 shows the Sub-band VQ-VAE quantization process. The original SP vector (1, 1205) is divided into two parts. The first part is the VQ-VAE lower band consisting of the lower band SP vector x with the shape (1, 684) as the input, and the lower part encoder networks produce a z-latent matrix $z_e(x)$ with the shape (1, 171, 64). The reshaped z-latent N vectors by D_{embed} is reconstructed from $z_e(x)$. The embedding space uses 9 bits, and the length of each prototype vector is D_{embed}. In the VQ process, 9 bits are assigned to each N vector. The quantized N vectors are reconstructed by the inverse VQ process, and those N vectors need to reshape back into the shape (1, 171, 64) to represent a quantized z-latent matrix $z_q(x)$. The output vector (1, 684) that looks like the lower band SP vector is reconstructed by using $z_q(x)$ as input to the decoder networks.

The second part of the divided SP vector is the VQ-VAE higher band consisting of the higher band SP vector x with shape (1, 341) as input to the higher part encoder networks. The encoder networks produce a z-latent matrix $z_e(x)$ with shape (1, 86, 64). Same as the VQ-VAE lower band, the $z_e(x)$ can be reshaped into N vectors by D_{embed} to represent the reshaped z-latent. The embedding space is represented by 9 bits with the length of D_{embed}. In the VQ process, 9 bits are assigned to each N vector. The quantized N vectors are reconstructed by the inverse VQ process and reshaped back into the shape (1, 86, 64) to produce a quantized z-latent matrix $z_q(x)$ The decoder uses $z_q(x)$

as the input to produce the output (1, 341), which corresponds to the higher band SP vector. The output of the VQ-VAE lower band and the VQ-VAE higher band are merged to reconstruct the output SP vector with the original shape (1, 1025).

The training of four VQ-VAE models used 0.0001 as the learning rate. Adam optimizer was applied to optimize network parameters. The number of training epochs was set to 500,000, and each epoch used the random of eight SP vectors from the SP vector database as a mini-batch. Figure 5.7 shows the VQ-VAE training process. In the training process, the encoder network received the input SP vector and produced the z-latent. The VQ process found the z-latent that minimized the Euclidian distance between the input and the vector patterns in the codebook. It returned the index of the vector pattern in the embedding space as the discrete representation of the z-latent. In the inverse VQ process, the obtained index picked the corresponding vector pattern in the embedding space to represent the quantized z-latent. The decoder network produced the quantized SP vector from the quantized z-latent. In the end, the loss of Equation 5.1 was calculated, and the Adam optimization was applied to update network parameters consisting of the encoder network, decoder network, and embedding space.

Figure 5.8 shows the proposed Sub-band VQ-VAE training process. The training of four models of the Sub-band VQ-VAE was also performed with the following conditions: the learning rate = 0.0001, network parameter optimization = Adam, the number of training epochs = 500,000, minibatch = the random of eight SP vectors from the SP vector database. In training, each encoder network for the lower and higher bands received the input SP vector (the lower band and the higher band) and produced the z-latent for the lower and higher bands. The VQ process found the z-latent which minimized the Euclidian distance with vector patterns in the designed embedding space. It returned the index of the vector pattern in the embedding space as the discreet representation of the z-latent. In the inverse VQ process, the obtained indices (the lower band and the higher band) chose the corresponding vector patterns in the embedding space to represent the quantized z-latent for the lower and higher band. Each decoder network of the lower and higher bands produced each quantized SP vector from the quantized z-latent. In the end, the loss of Equation 5.1 was calculated, and the Adam optimizer updated the network parameters consisting of the encoder network, decoder network, and embedding space of the lower band and higher band.

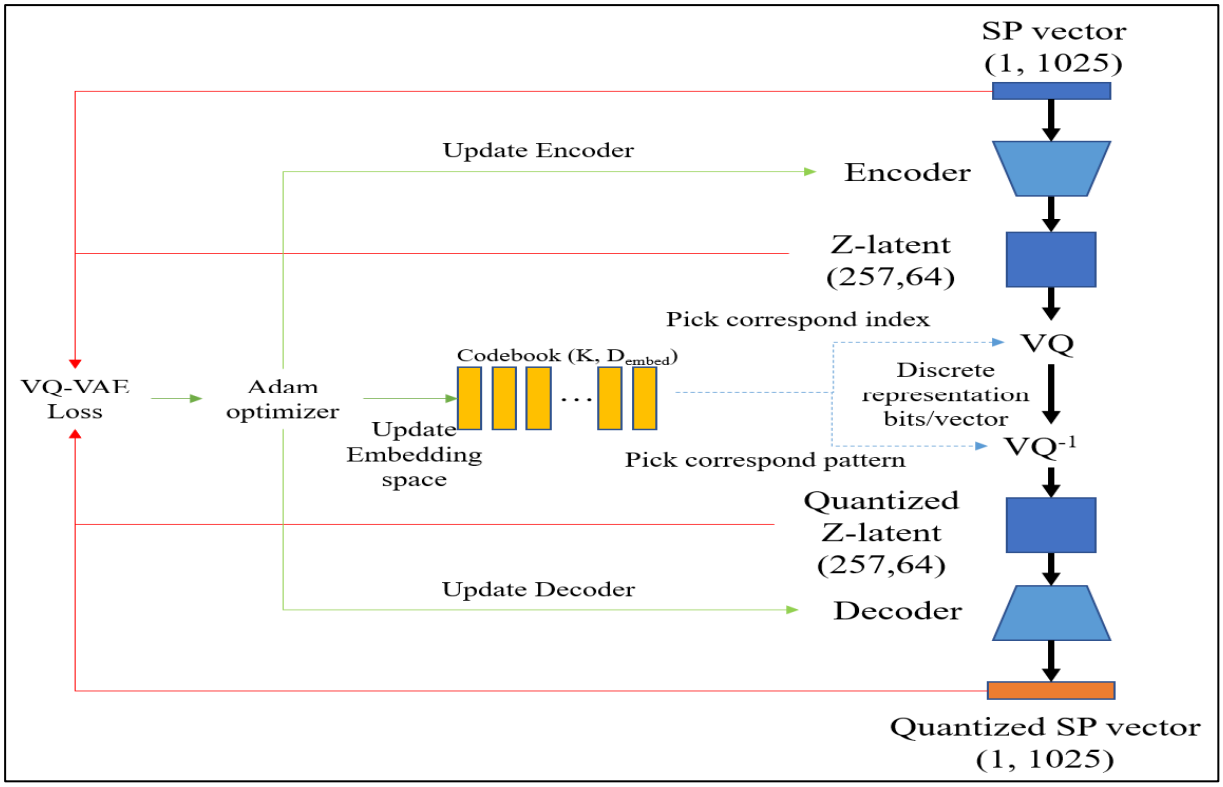

Figure 5.7: The VQ-VAE training process.

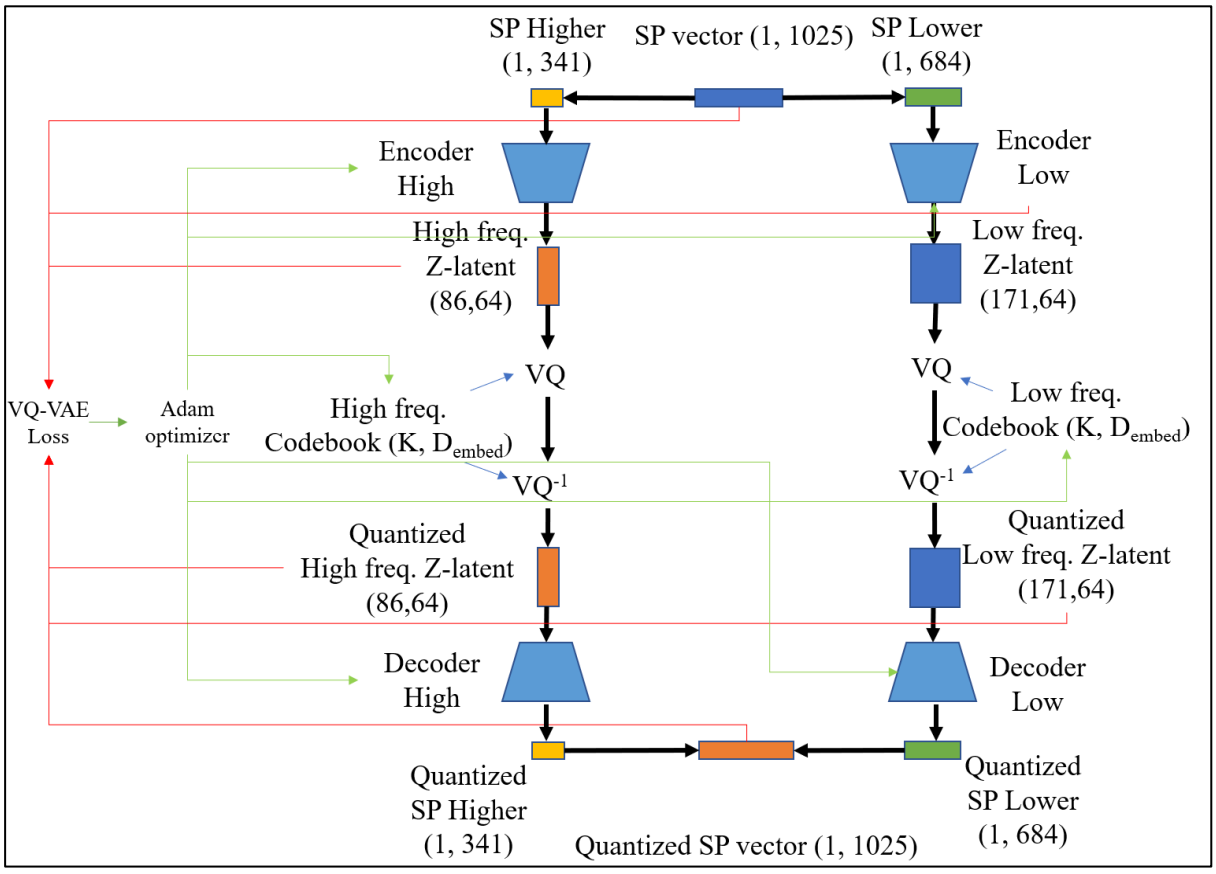

Figure 5.8: The Sub-band VQ-VAE training process.

Table 5.2
The Implemented four model comparison.

VQ technique	Shape of z-latent (H, W, D)	Shape of reshaped z-latent (N, D_{embed})	Shape of embedding space (K, D_{embed})
VQ-VAE	(1, 257, 64)	(8, 2056)	(512, 2056)
		(32, 514)	(512, 514)
		(64, 257)	(512, 257)
		(256, 64)	(512, 64)
sub-band VQ-VAE	Low freq. (1, 171, 64)	Low freq. (6, 1824) + High freq. (2, 2752)	Low freq. (512, 1824) + High freq. (512, 2752)
		Low freq. (24, 456) + High freq. (8, 688)	Low freq. (512, 456) + High freq. (512, 688)
	High freq. (1, 86, 64)	Low freq. (48, 228) + High freq. (16, 344)	Low freq. (512, 228) + High freq. (512, 344)
		Low freq. (171, 64) + High freq. (86, 64)	Low freq. (512, 64) + High freq. (512, 64)

Table 5.2 shows the detailed implementation for the four models of the Sub-band VQ-VAE and the VQ-VAE. The comparison of embedding space is also shown in the table. The sub-band VQ-VAE uses a larger embedding space size than the VQ-VAE. Two independent VQ-VAE models are used to quantize the lower band SP vector and the higher band SP vector.

As an evaluation of the performance, we use the Log Spectral Distortion (LSD) [85, 86] to measure spectral distortion between the original and reconstructed spectral envelopes. The LSD is defined as follows:

$$LSD_{(dB)} = 10 \times \frac{2}{M}\sqrt{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left(X_{ij} - Y_{ij}\right)^2}, \tag{5.2}$$

where M is the number of log-spectral coefficient frames, N is the length of log-spectral coefficient elements in a frame, X_{ij} is the original log10 spectral

coefficients of the WORLD parameters, and Y_{ij} is the log10 spectral coefficients of the WORLD reconstructed parameters from the quantized data.

Figures 5.9 and 5.10 calculate the average LSD evaluation processes for the VQ-VAE and the proposed Sub-band VQ-VAE, respectively. The testing set consists of 100 raw speech waveforms from the VCTK corpus that are not included in the training process of the VQ-VAE and the proposed Sub-band VQ-VAE. The WORLD vocoder extracts the speech parameters from each raw speech waveform, only the spectral envelope parameter (SP) is applied to quantize, and the synthesis process reconstructs the raw speech waveform again based on the quantized SP and other parameters. The LSD is calculated to measure the distortion between the SP and the quantized SP for each time feeding raw speech waveforms. Finally, the average LSDs from 100 LSD values are calculated from 100 testing speech sentences.

The evaluated results are obtained by applying the quantization technique to the SP vector of a high-quality WORLD vocoder. As the input to the encoder, the SP vector is needed to be normalized into the value between 0 to 1 by min-max normalization. Furthermore, the reconstructed SP vector obtained from the quantization techniques is processed by inverse min-max normalization to recover the real values as the WORLD synthesis part parameters. In the process of designed experiments, the reshaped z-latent has a finite number to control the number of N vectors because each vector needs to have the same length D_{embed}.

We evaluated four reshaped z-latent targets; the number of N vectors are 257, 64, 32, and 8 vectors. From the four targets, the number of reshaped z-latent N vectors changes the D_{embed} length. When the number of N vectors is large, the D_{embed} length is short. On the other hand, when the number of N vectors is small, the D_{embed} length is long. The number of K vectors for embedding space is set to 512 (9 bits). The four reshaped z-latent targets have the number of N vectors of 257, 64, 32, and 8, where each vector is assigned with 9 bits. The four target bitrates were 2313, 576, 288, and 72 bits/SP vector, respectively.

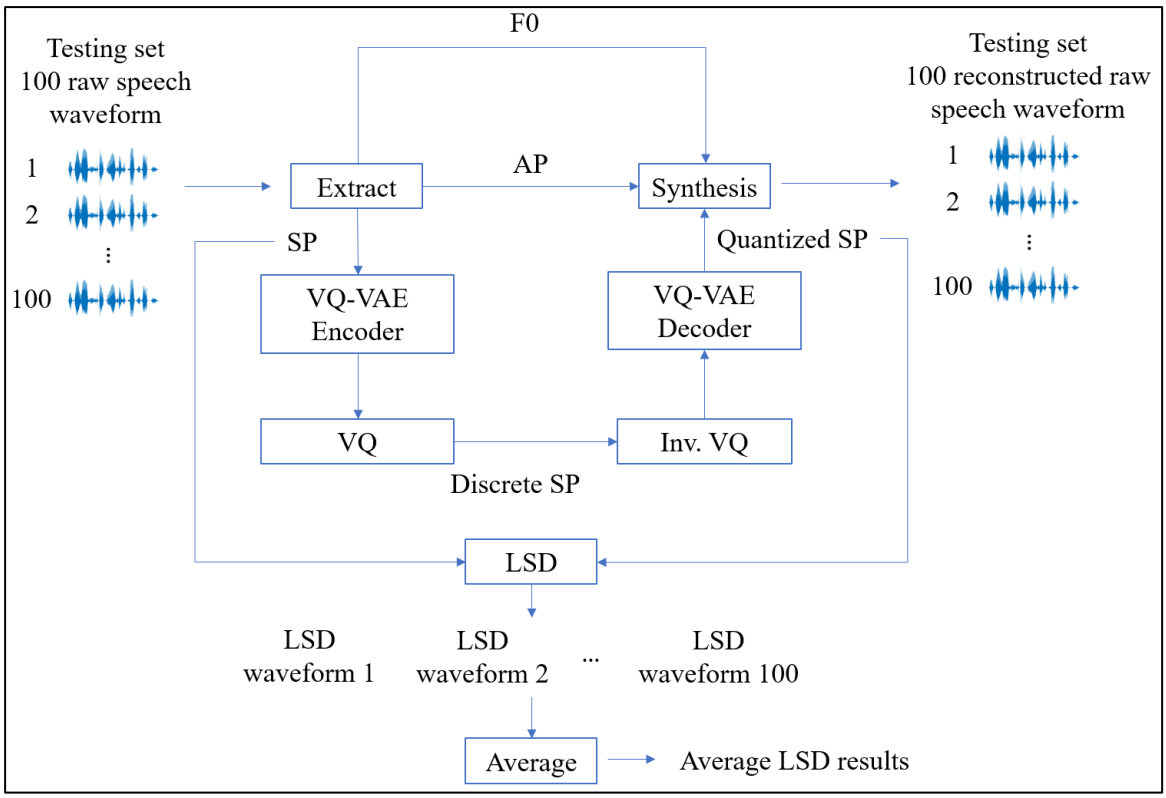

Figure 5.9: The VQ-VAE average LSD evaluation.

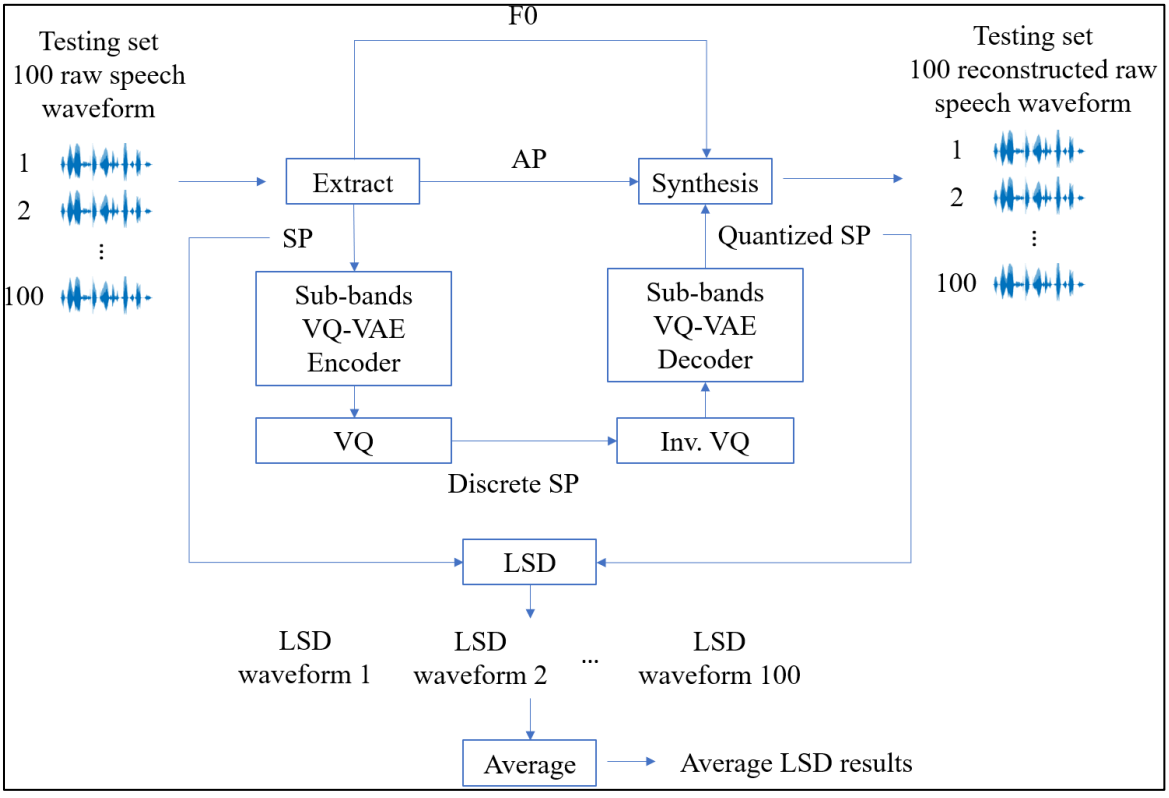

Figure 5.10: The proposed Sub-band VQ-VAE average LSD evaluation.

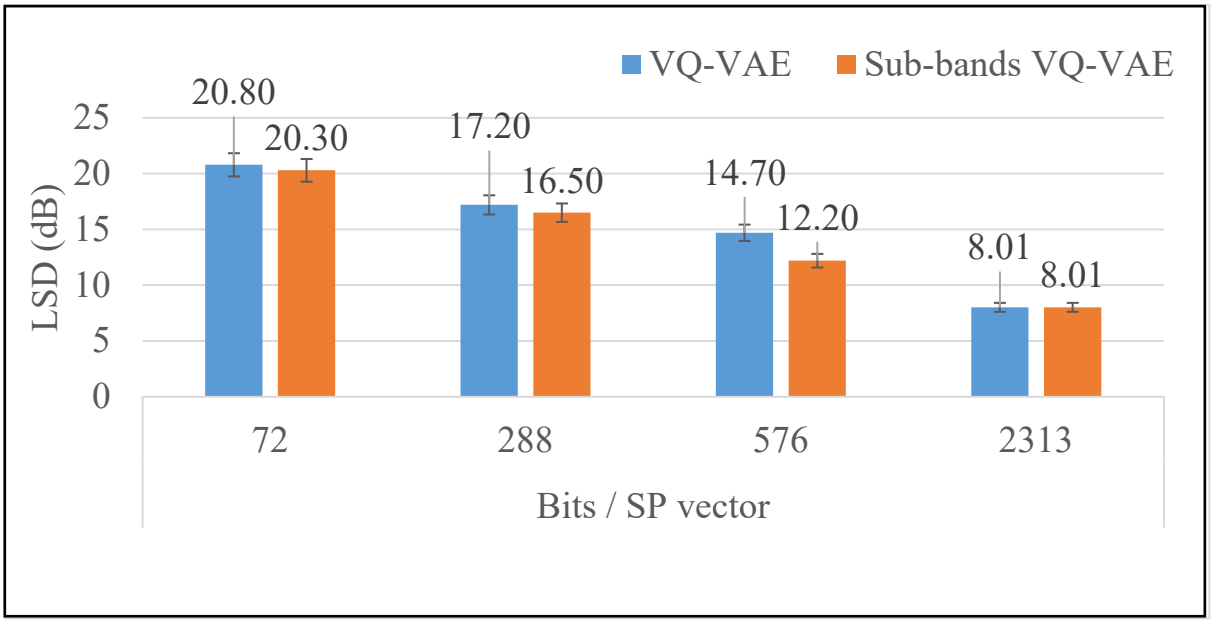

Figure 5.11: The comparison of LSD (in dB) in four target bitrates.

Figure 5.11 shows the comparison of LSD with four target bitrates has shown in. In the low bitrates, the LSD is high, and it decreases when the bitrate increases. The results show that the proposed Sub-band VQ-VAE can decrease the LSD more than the VQ-VAE.

5.3.4 The Predictive Vector Quantized Variational AutoEncoder for Spectral Envelope Quantization

The VQ-VAE and the Predictive VQ-VAE were designed for quantization of the spectral envelope parameters extracted from the high-quality 48 kHz WORLD vocoder. The WORLD vocoder extracted speech parameters, the fundamental frequency (F0), the spectral envelope parameter (SP), and the aperiodic parameter (AP), at every 5 ms from the raw 48 kHz sampling speech waveform. The reconstruction process appropriated the F0, SP, and AP speech parameters to provide the 5 ms output of 48 kHz speech waveform by synthesis. SP parameter was the essential speech parameter that contains phonemic information.

The VQ-VAE architecture for the comparison of the Predictive VQ-VAE is shown in Table 5.3. The encoder network was implemented as the two stride convolutional layers connected with two residual networks and followed the one convolutional layer. The decoding network was the counterpart of the encoder network, utilized the transposed convolutional network. The decoder network was implemented as the one transposed convolution layer combined

with the two residual networks and supported the two transposed convolutional layers.

Table 5.4 shows the Predictive VQ-VAE architecture. The model consists of four networks, the encoder network, the encoder predictor network, the decoder network, and the decoder predictor network. The encoder network was implemented as two stride convolutional layers with the two residual networks following the one convolutional network. The encoder predictor network was also implemented equivalent to the original encoder network, but the weight filter shapes were changed. The decoder network was implemented as one transposed convolutional layer attached to the two residual networks, which transposed the convolutional layers. The decoder predictor network was also the same implementation of the decoder network, but the weight filter shapes were different.

Table 5.3
The VQ-VAE architecture for the comparison of the Predictive VQ-VAE.

Network type	Architecture	Weight (Height, Width, Depth)	Bias	Stride (Height, Width)	Activation function
Encoder Networks	CNN input layer 1	(4, 4, 64)	64	(2,2)	ReLU
	CNN input layer 2	(4, 4, 128)	128	(2,2)	ReLU
	CNN Res. I layer 1	(3, 3, 64)	64	(1,1)	ReLU
	CNN Res. I layer 2	(1, 1, 128)	128	(1,1)	ReLU
	CNN Res. II layer 1	(3, 3, 64)	64	(1,1)	ReLU
	CNN Res. II layer 2	(1, 1, 128)	128	(1,1)	ReLU
	CNN prepare VQ layer	(1, 1, 64)	64	(1,1)	-
Decoder Networks	Transposed CNN Inv. Prepare VQ layer	(1, 1, 128)	128	(1,1)	ReLU
	CNN Res. I layer 1	(3, 3, 64)	64	(1,1)	ReLU
	CNN Res. I layer 2	(1, 1, 128)	128	(1,1)	ReLU
	CNN Res. II layer 1	(3, 3, 64)	64	(1,1)	ReLU
	CNN Res. II layer 2	(1, 1, 128)	128	(1,1)	ReLU
	Transposed CNN output layer 1	(4, 4, 64)	64	(2,2)	ReLU
	Transposed CNN output layer 2	(4, 4, 1)	1	(2,2)	Sigmoid

Table 5.4
The Predictive VQ-VAE architecture.

Network type	Architecture	Weight (Height, Width, Depth)	Bias	Stride (Height, Width)	Activation function
Encoder Networks	CNN input layer 1	(4, 4, 64)	64	(2,2)	ReLU
	CNN input layer 2	(4, 4, 128)	128	(2,2)	ReLU
	CNN Res. I layer 1	(3, 3, 64)	64	(1,1)	ReLU
	CNN Res. I layer 2	(1, 1, 128)	128	(1,1)	ReLU
	CNN Res. II layer 1	(3, 3, 64)	64	(1,1)	ReLU
	CNN Res. II layer 2	(1, 1, 128)	128	(1,1)	ReLU
	CNN prepare VQ layer	(1, 1, 64)	64	(1,1)	-
Encoder Predictor Networks	CNN input layer 1	(4, 4, 128)	128	(1,1)	ReLU
	CNN input layer 2	(4, 4, 256)	256	(1,1)	ReLU
	CNN Res. I layer 1	(3, 3, 128)	128	(1,1)	ReLU
	CNN Res. I layer 2	(1, 1, 256)	256	(1,1)	ReLU
	CNN Res. II layer 1	(3, 3, 128)	128	(1,1)	ReLU
	CNN Res. II layer 2	(1, 1, 256)	256	(1,1)	ReLU
	CNN prepare VQ layer	(1, 1, 64)	64	(1,1)	-
Decoder Networks	Transposed CNN Inv. Prepare VQ layer	(1, 1, 128)	128	(1,1)	ReLU
	CNN Res. I layer 1	(3, 3, 64)	64	(1,1)	ReLU
	CNN Res. I layer 2	(1, 1, 128)	128	(1,1)	ReLU
	CNN Res. II layer 1	(3, 3, 64)	64	(1,1)	ReLU
	CNN Res. II layer 2	(1, 1, 128)	128	(1,1)	ReLU
	Transposed CNN output layer 1	(4, 4, 64)	64	(2,2)	ReLU
	Transposed CNN output layer 2	(4, 4, 1)	1	(2,2)	Sigmoid
Decoder Predictor Networks	Transposed CNN Inv. Prepare VQ layer	(1, 1, 128)	128	(1,1)	ReLU
	CNN Res. I layer 1	(3, 3, 64)	64	(1,1)	ReLU
	CNN Res. I layer 2	(1, 1, 128)	128	(1,1)	ReLU
	CNN Res. II layer 1	(3, 3, 64)	64	(1,1)	ReLU
	CNN Res. II layer 2	(1, 1, 128)	128	(1,1)	ReLU
	Transposed CNN output layer 1	(4, 4, 64)	64	(1,1)	ReLU
	Transposed CNN output layer 2	(4, 4, 64)	64	(1,1)	-

Figure 5.12 shows the quantization process of the VQ-VAE. The SP vector is the input of the encoder network to produce the z-latent matrix. The reshaping method was applied to the z-latent matrix for the reshaped z-latent vectors with the vector length corresponding to the vector length of the designed embedding space. The VQ was applied to transform the continuously reshaped z-latent vectors into a discrete presentation, and the inverse VQ was transformed back to the quantized continuous z-latent matrix as quantized reshaped z-latent vectors. The input of the decoder network was the quantized z-latent matrix reshaped from the quantized reshaped z-latent vectors and reproduces the quantized SP vector.

In the training of VQ-VAE models, the SP vector database was applied to the logarithmic base ten and min-max normalization to transform the values into a scale between 0 to 1. The learning threshold was set to 0.0002. The training number was 100,000 epochs, and each epoch processed the 8 SP vectors as mini-batch. Adam optimization was applied to optimize the networks. Figure 5.13 shows the VQ-VAE training process. In training, the encoder network received the input SP vector and produces the z-latent. The VQ process utilized the z-latent to find the minimum Euclidian distance. It returns the index of the vector pattern in the embedding space as the discreet representation of the z-latent. In the inverse VQ process, the obtained index is utilized to choose the corresponded vector pattern in the embedding space to represent the quantized z-latent. The decoder network produces the quantized SP vector from the quantized z-latent. In the end, the loss of Equation 5.1 was calculated, and the Adam optimizer updated network parameters consisting of the encoder network, decoder network, and embedding space.

Figure 5.14 shows the quantization process of the Predictive VQ-VAE. The SP Vector was fed into the encoder network in the encoding process to produce the z-latent matrix. It was reshaped in N sub-vectors corresponding to the designed embedding space (K= 2^n $bits$, D_{embed}). The quantization process created the discrete representation, and the encoder predictor network received the quantized z-latent as the input. The predicted z-latent matrix was processed to subtract from the next z-latent matrix and add to the next quantized z-latent. In the decoding process, the received discrete representation reproduced the quantized z-latent. The decoder predictor network employed the quantized z-latent as the input to produce the predicted z-latent matrix from the decoder predictor network to add with the next quantized z-latent. The decoder network utilized the current quantized z-latent to reconstruct the quantized SP vector indicated to the SP vector.

Figure 5.15 shows the Predictive VQ-VAE training process. For the training of Predictive VQ-VAE models, like the training of VQ-VAE models, the SP vector database was applied the logarithmic base 10 and min-max normalization to transform the values into a scale between 0 to 1. The learning threshold was set to 0.0002. The training number was 100,000 epochs, and each epoch processed the 8 SP vectors as mini-batch. Adam optimization was applied to optimize the networks. In the training process, the SP vector is applied to the Predictive VQ-VAE process to obtain the quantized SP vector, z-latent, quantized z-latent. In the end, the VQ-VAE loss based on Equation 5.1 was calculated and the Adam optimizer updated the network parameters consisted of the encoder network, decoder network, embedding space, encoder predictor network, and the decoder predictor network.

In the experiment, the number of vector patterns K of the embedding space was fixed to 8 bits or 256 vector patterns. The four VQ-VAE models were designed for spectral envelope parameter quantization, presented in Table 5.5. The shape of the input SP parameter was (H=1, W=1025, D=1), where H was height, W was width, and D was depth dimension. The encoder network produced the z-latent $z_e(x)$ with the shape (H=1, W=257, D=64). The reshaped $z_e(x)$ was the N sub-vectors by the length of D_{embed} (corresponded to the length of embedding space). The reshaped $z_e(x)$ had limited shape because of the limitation of the reshaping method. In this experiment, the four targets $(N, D_{embed}) =$ (8, 2056), (16, 1028), (32, 514), and (64, 257) were selected, and the final bits/SP vectors were the 64, 128, 256, and 512 for the first, second, third, and fourth VQ-VAE models, respectively.

The Predictive VQ-VAE also created four bitrate models to quantize the spectral envelope parameters, as shown in Table 5.6. The shape of the input SP vector was (H=1, W=1025, D=1). The encoder network created the z-latent $z_e(x)$, and the encoder predictor and decoder predictor networks created the $\tilde{z}_q(x)$ with the same shape (H=1, W=257, D=64). As for the reshaped $z_e(x)$, the same four targets as the four VQ-VAE models, $(N, D_{embed}) =$ (8, 2056), (16, 1028), (32, 514), and (64, 257) were selected, and the final bits/SP vector were the 64, 128, 256, and 512 bits/SP vector for the first, second, third, and fourth Predictive VQ-VAE models, respectively.

Table 5.5
The four VQ-VAE implementation models
for the comparison with Predictive VQ-VAE.

Model	Input SP	$z_e(x)$	reshaped $z_e(x)$		Embedding space		$\frac{Bits}{SP\ vector}$
	(H, W, D)	(H, W, D)	N	D_{embed}	K	D_{embed}	
VQ-VAE 1	(1, 1025, 1)	(1, 257, 64)	8	2056	256	2056	64
VQ-VAE 2			32	514		514	256
VQ-VAE 3			64	257		257	512
VQ-VAE 4			257	64		64	2056

Table 5.6
The four Predictive VQ-VAE implementation models.

Model	Input SP	$z_e(x)$	$\tilde{z}_q(x)$	reshaped $z_e(x)$		Embedding space		$\frac{Bits}{SP\ vector}$
	(H, W, D)	(H, W, D)	(H, W, D)	N	D_{embed}	K	D_{embed}	
Predictive VQ-VAE 1	(1, 1025, 1)	(1, 257, 64)	(1, 257, 64)	8	2056	256	2056	64
Predictive VQ-VAE 2				32	514		514	256
Predictive VQ-VAE 3				64	257		257	512
Predictive VQ-VAE 4				257	64		64	2056

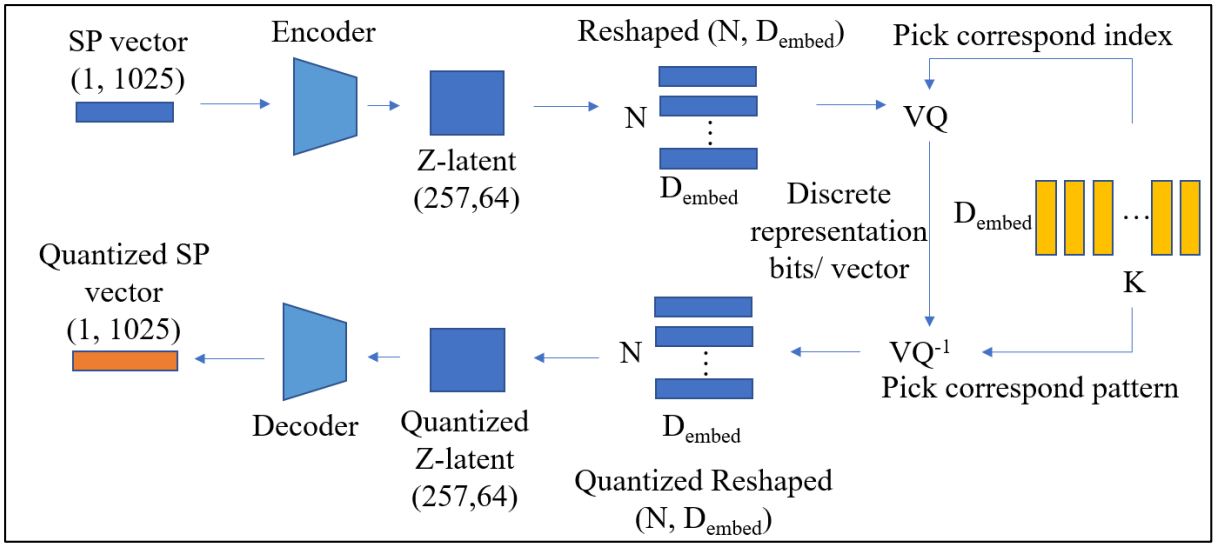

Figure 5.12: The VQ-VAE quantization process for the comparison with the Predictive VQ-VAE.

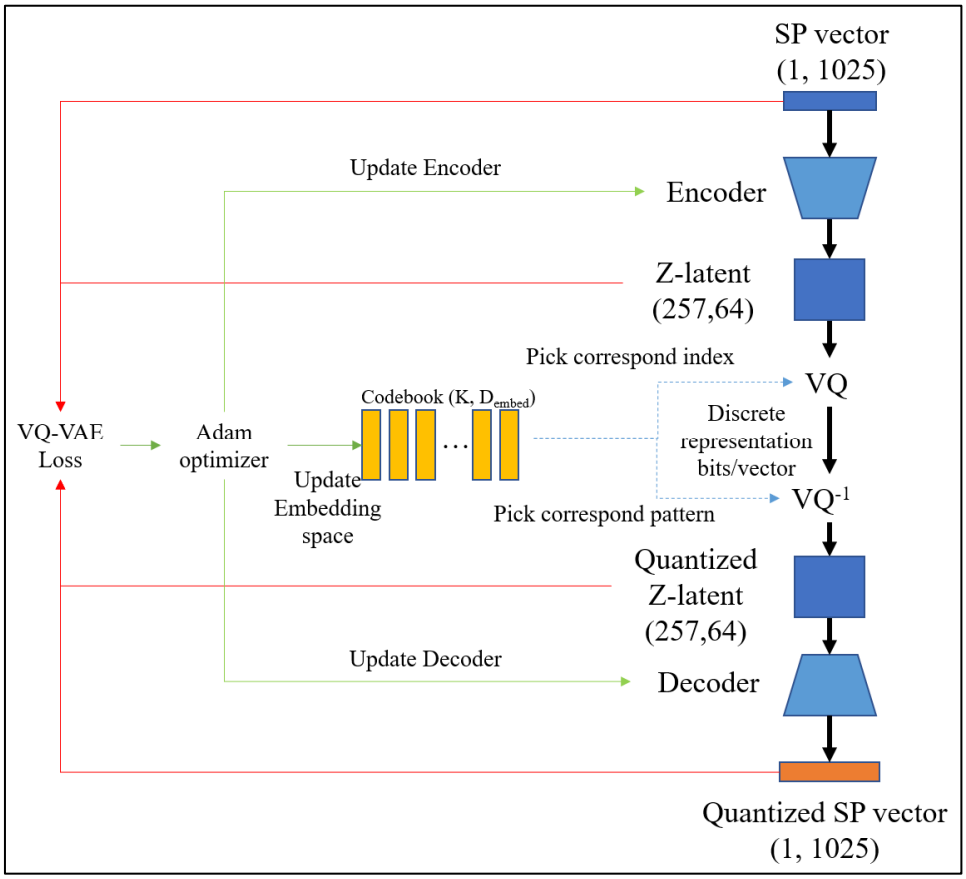

Figure 5.13: The VQ-VAE training process for the comparison with the Predictive VQ-VAE.

.

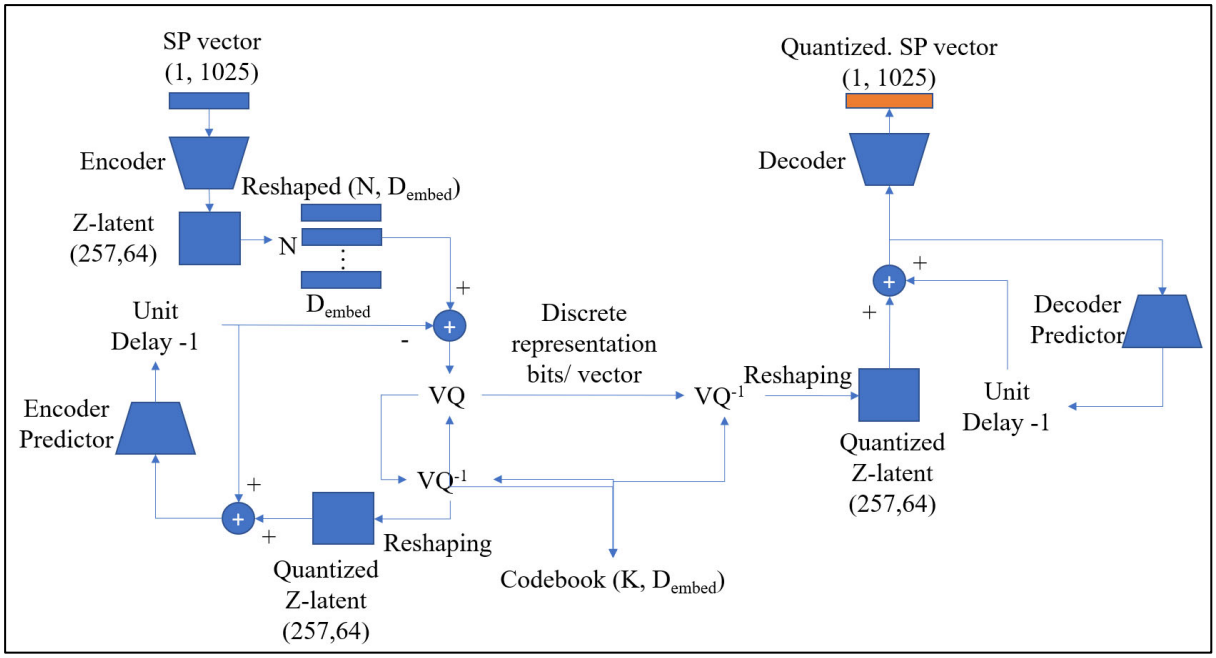

Figure 5.14: The Predictive VQ-VAE quantization process.

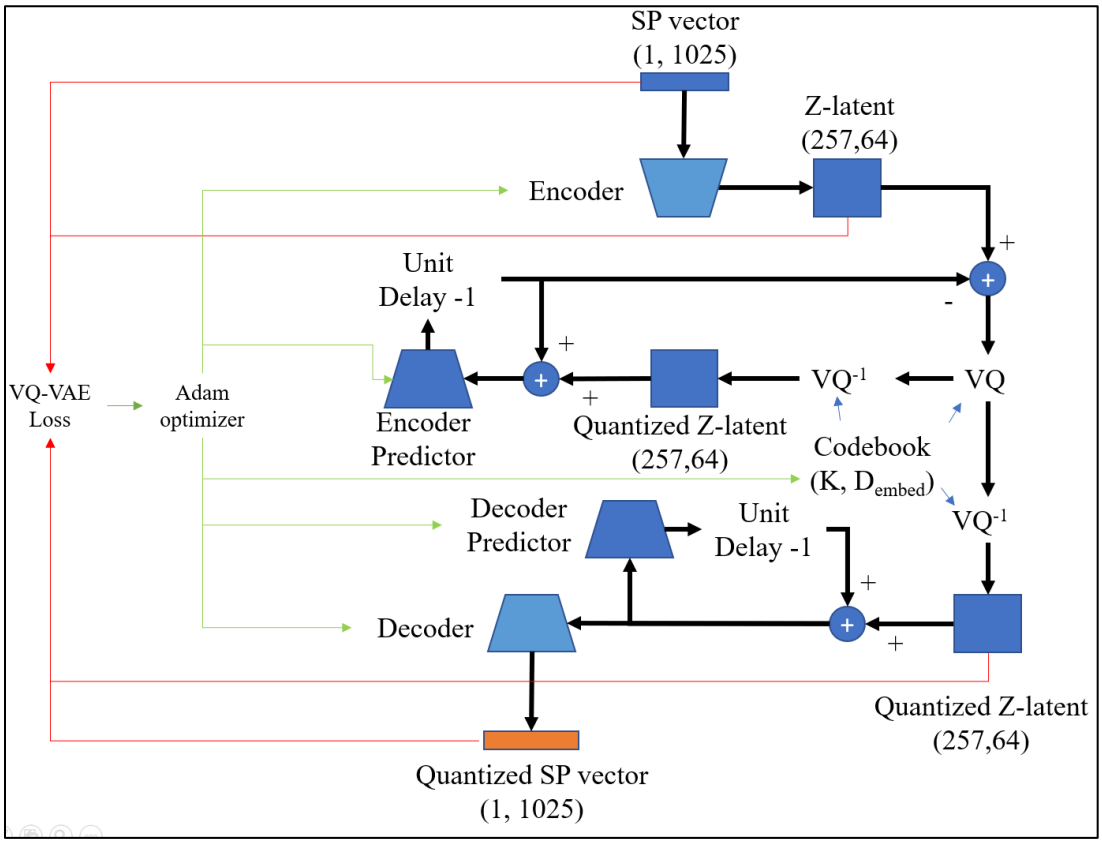

Figure 5.15: The Predictive VQ-VAE training process.

The results were evaluated by using 100 test speech waveforms. The four VQ-VAE models and the four Predictive VQ-VAE models were applied to the WORLD vocoder to quantize only the SP parameter. The vocoder extracted speech parameters from input five milliseconds of raw speech waveform and calculated F0, SP, and AP parameters. The SP parameter was applied to the logarithmic base ten and min-max normalization to transform the values into the scale between 0 to 1, and the quantization technique was implemented to quantize the normalized SP vector. The synthesis process of the WORLD vocoder received the F0, quantized SP, and AP to reconstruct the output 5 ms speech waveform. The Log spectral distortion (LSD) was used to measure the distortion shown in Equation 5.2.

The experimental results utilized the Log Spectral Distortion (LSD) [85, 86] in Equation 5.2 as the spectral envelope distortion indicator. Figures 5.16 and 5.17 show the process of calculating the average LSDs for the VQ-VAE and the proposed Predictive VQ-VAE, respectively. The testing set consisted of 100 raw speech waveforms from the VCTK corpus that were not included in the training process of the VQ-VAE and the proposed Predictive VQ-VAE. The WORLD vocoder extracted the speech parameters from each raw speech waveform, only the spectral envelope parameter (SP) was quantized, and the synthesis process reconstructed the raw speech waveform again based on the quantized SP and other parameters. The LSD between SP and quantized SP for each time that feeding raw speech waveform was calculated. Finally, the average LSD is obtained from 100 tested LSD values.

Figure 5.18 shows the LSD comparison of the four VQ-VAE models and the four Predictive VQ-VAE models with four targets of bits/SP vector. Figure 5.19 compares the spectrograms of the VQ-VAE and the Predictive VQ-VAE SP vectors. The average LSD results decreased when the bits/SP vector increased. The Predictive VQ-VAE decreased the average LSD at the 2,056 bits/SP vector at around 5.7 dB, at the 517 bits/ SP vector at around 2.2 dB, at the 257 bits/ SP vector at around 1.95 dB, and at the 64 bits/ SP vector around 0.4 dB, compared to the four VQ-VAE models.

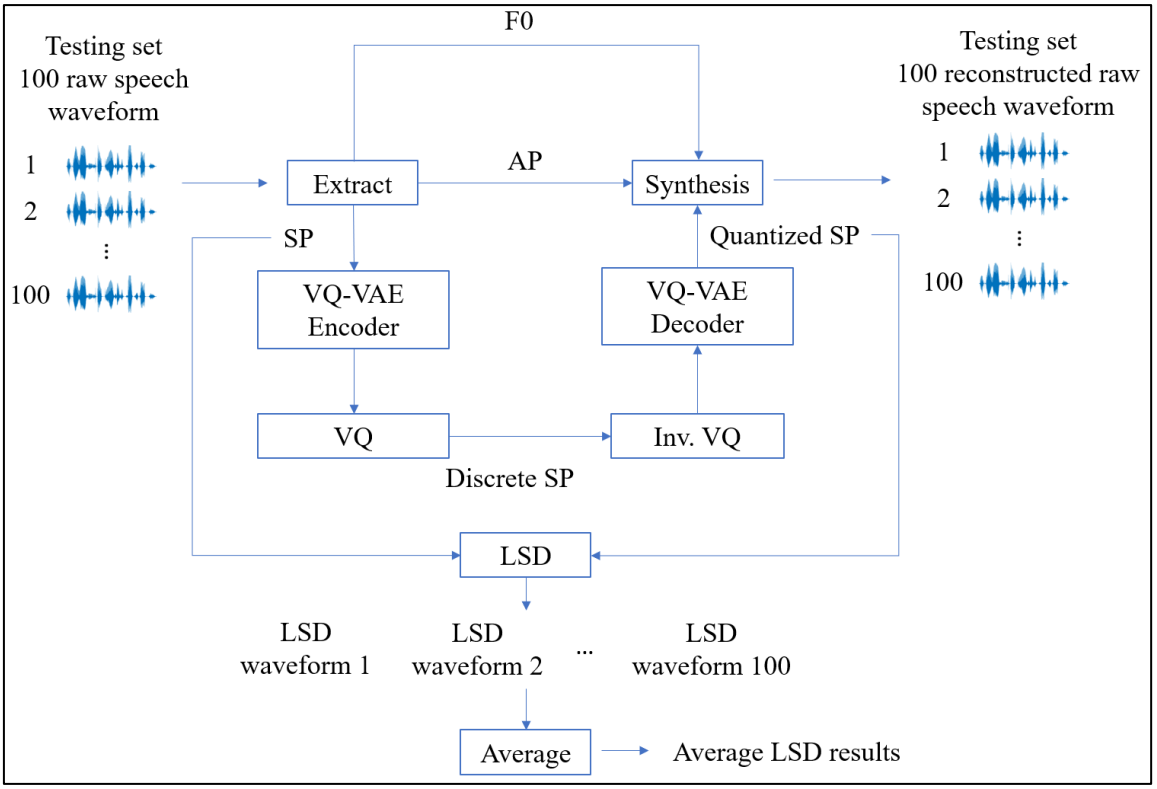

Figure 5.16: The VQ-VAE average LSD evaluation.

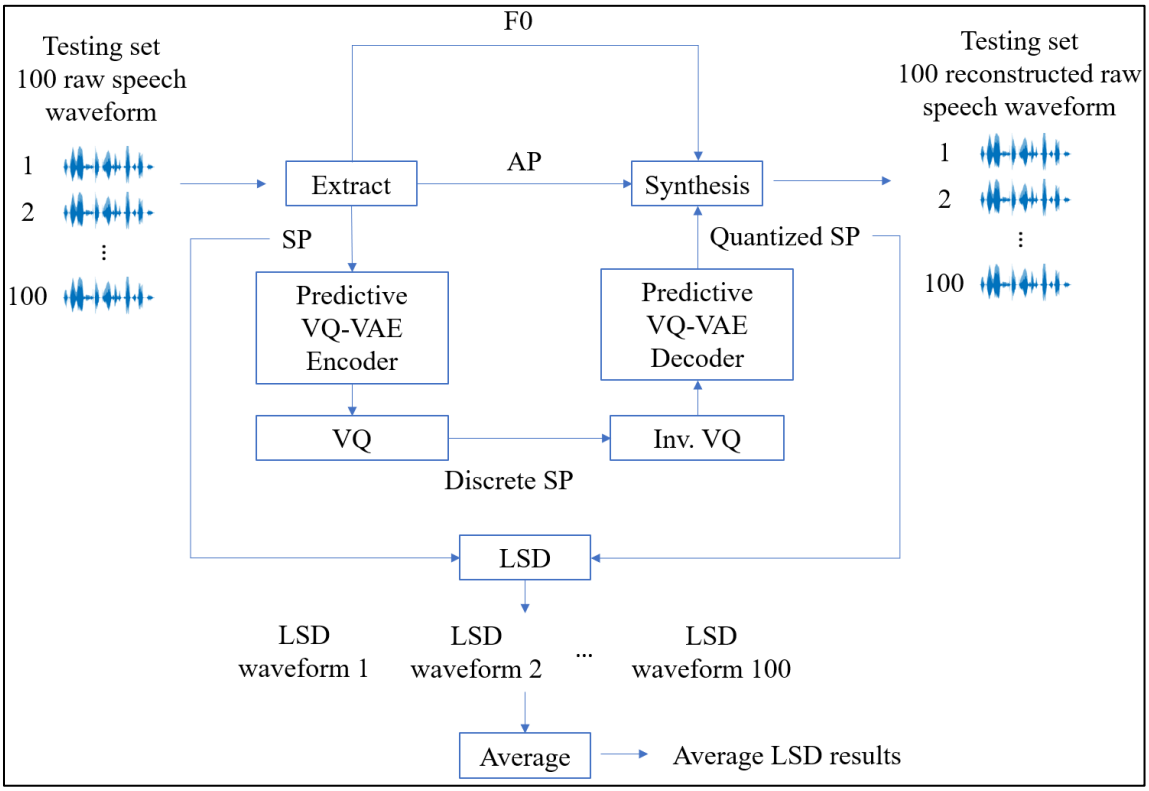

Figure 5.17: The proposed Predictive VQ-VAE average LSD evaluation.

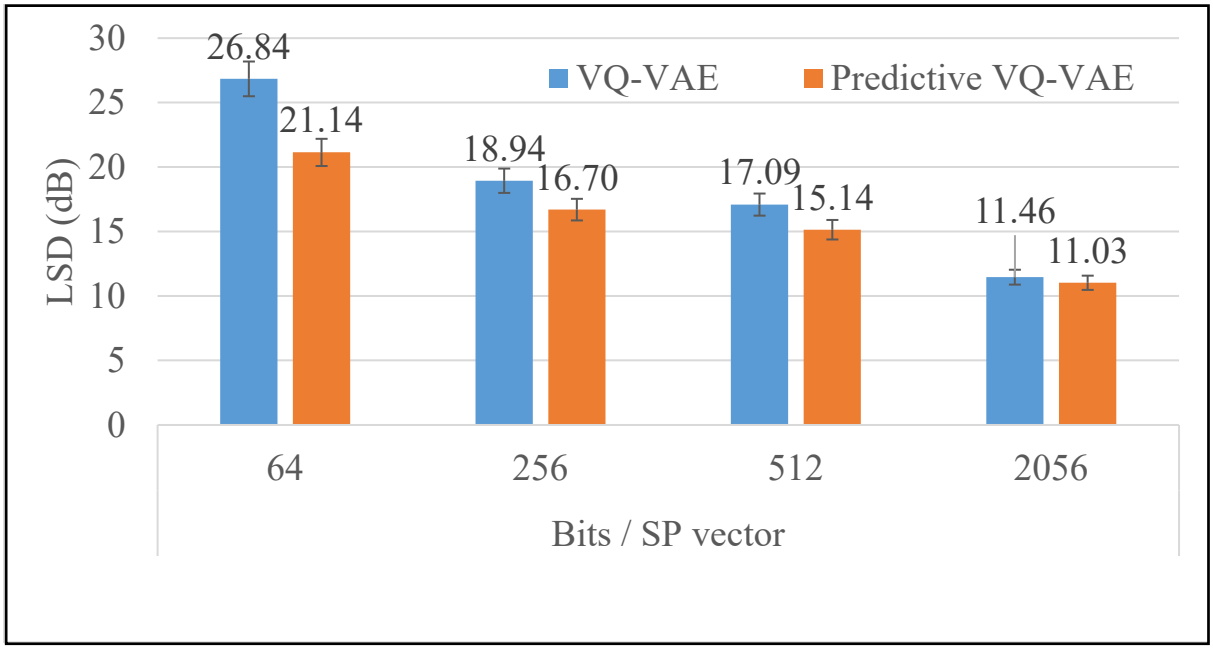

Figure 5.18: The comparison of average LSD results in four target bitrates.

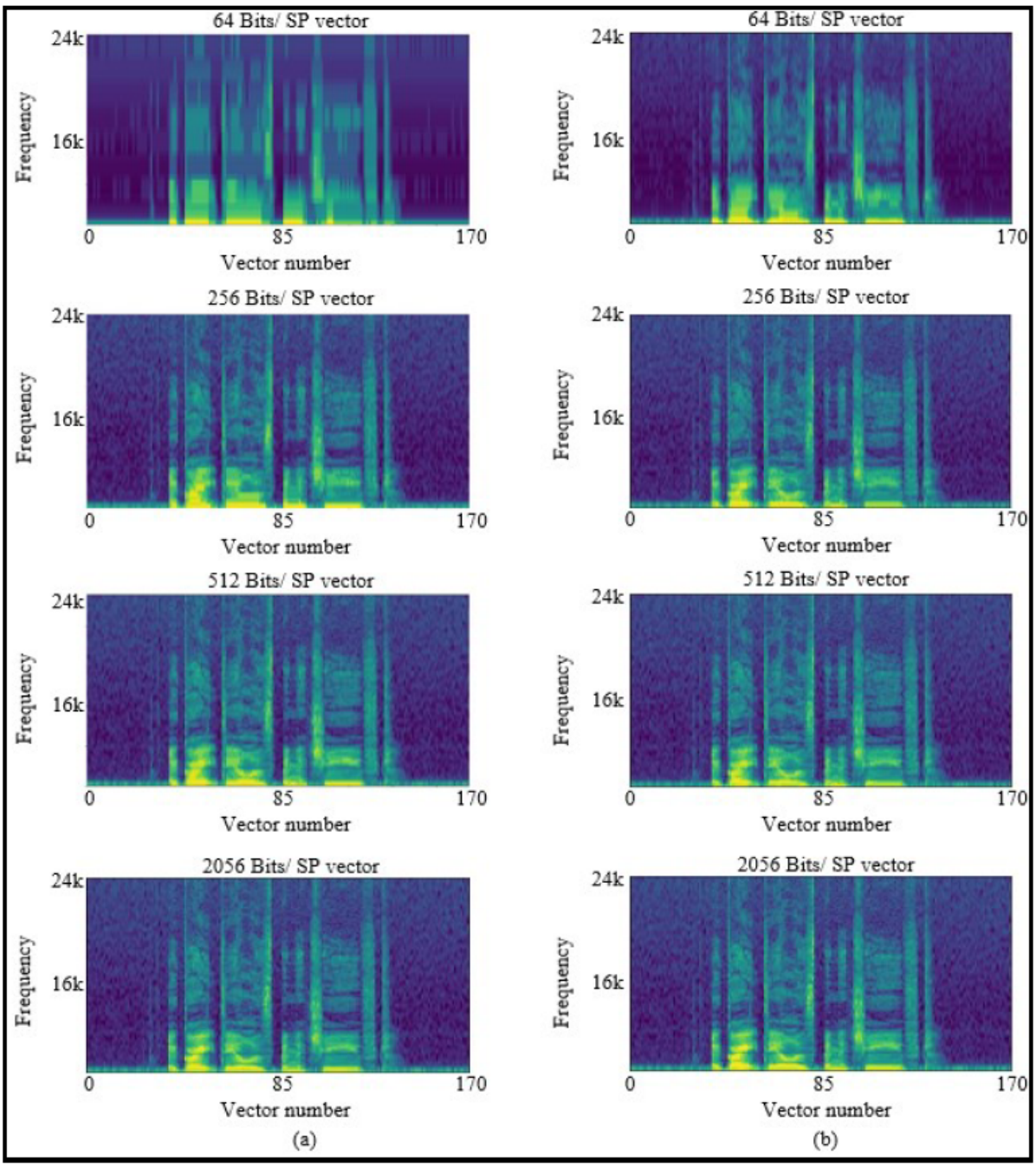

Figure 5.19: The comparison of quantized SP vectors in four target bitrates; (a) is the VQ-VAE and (b) is the Predictive VQ-VAE.

5.4 Discussion

5.4.1 The Sub-band Vector Quantized Variational AutoEncoder for Spectral Envelope Quantization

The sub-band coding technique was investigated to work with VQ-VAE. The VQ-VAE is an end-to-end deep learning method based on the VQ technique. We quantized the spectral envelope parameters of the WORLD vocoder to evaluate the results. The SP vector was divided into two sub-band frequencies. The lower band frequency is more significant than the higher one because most of the speech information is kept inside it, and the higher band frequency has less speech information.

The quantization of VQ-VAE using the full length of the SP vector could not concentrate on a specific frequency band. To solve this problem, the Sub-band VQ-VAE was proposed to design the model for sub-band frequency quantization. This model could assign more bits to the lower frequency band and fewer bits to the unnecessary higher frequency band. The experimental performance results showed that Sub-band VQ-VAE was successful with the lower LSD distortion in four different bitrates around 0.93 dB on average. However, because the SP vector was split into two sub-vectors in two independent embedding spaces, the VQ-VAE was applied to each sub-vector for quantization. The sub-band VQ-VAE needs around 2.17 times more embedding space than the single-band VQ-VAE.

5.4.2 The Predictive Vector Quantized Variational AutoEncoder for Spectral Envelope Quantization

In this section, the Predictive coding technique was investigated to work with VQ-VAE. The VQ-VAE could not produce the output from the previous information of the input. Therefore, we introduced the predictive coding technique, the Predictive VQ-VAE, in the VQ-VAE for utilizing the information of the previous data to produce the current output data.

The experimental results the performance showed that Predictive VQ-VAE was successful with the lower LSD distortion in four different bit rates, at the 2,056 bits/SP vector around 5.7 dB, at the 517 bits/ SP vector around 2.2 dB, at the 257 bits/ SP vector around 1.95 dB, and at the 64 bits/ SP vector around 0.4 dB, compared to the corresponding VQ-VAE models. However, the model complexity increased a lot because the Predictive VQ-VAE required

the encoder predictor network and the decoder predictor network which made the model has time-consuming more than the VQ-VAE.

5.5 Conclusion

In conclusion, this chapter provides the following contributions:

- The sub-band VQ-VAE was proposed to quantize the spectral envelope parameters of the high-quality 48kHz WORLD vocoder. This model can focus on a specific sub-band frequency by assigning more quantization bits and leaving unnecessary sub-band frequencies with fewer bits.

- The Sub-band VQ-VAE estimated the performance for the quantization of the spectral envelope parameter of a high-quality WORLD vocoder that operates at 48kHz raw speech waveform. The LSD results showed that the average results from four operation bitrates of the sub-band VQ-VAE had lower LSD values than the VQ-VAE, around 0.93 dB.

- Since the Sub-band VQ-VAE required the embedding space more substantial than the VQ-VAE, around 2.17 times, the effective representation of the codebooks is a future problem.

- The Predictive VQ-VAE was proposed to quantize the spectral envelope parameters of the high-quality 48kHz WORLD vocoder and inspired by the predictive vector quantization, and the quantization technique can utilize the previous data to produce the current data.

- The Predictive VQ-VAE showed good performance for the quantization of the spectral envelope parameter of a high-quality WORLD vocoder that operates for 48kHz speech waveforms. It was shown that the Predictive VQ-VAE had a lower distortion in terms of Log Spectral Distortion for four targets bitrates associated with the VQ-VAE, at the 2,056 bits/SP vector around 5.7 dB, at the 517 bits/ SP vector around 2.2 dB, at the 257 bits/ SP vector around 1.95 dB, and at the 64 bits/ SP vector around 0.4 dB, compared to the four VQ-VAE models. The LSD results showed that the average results from four operation bitrates of the Predictive VQ-VAE had lower LSD values than the VQ-VAE, around 2.58 points in dB. The future problem is the model complexity, which increases a lot because the Predictive VQ-VAE requires both the encoder and the decoder predictor networks making the model time-consuming more than the VQ-VAE.

Chapter 6

The effect of deep learning network architecture and training techniques for speech spectral envelope quantization

6.1 Overview

The last study in the dissertation is the advanced deep learning training technique in VQ-VAE [44, 45], the combination between VQ-VAE and the Generative Adversarial Network [68] designed to work together in the spectral envelope quantization in four different target bitrates compared to the conventional VQ. The studies consist of the effect of the adversarial loss update on the whole networks of VQ-VAE and only the embedding space of quantization in the VQ-VAE.

We proposed the Vector Quantized Variational AutoEncoder learned by Generative Adversarial Networks and introduced an objective distortion major and training procedure of the GAN technique to replace the conventional distortion major of VQ-VAE. The experiment constructed four models for the following methods:

(1) the four VQ-VAE models, (references)
(2) the four VAEGAN [73] implemented in VQ-VAE models,
(3) the four VQ-VAE-EMGAN models, and
(4) the four VQ-VAE-EMDEC models.

Those models were designed and trained for the quantization of the spectral envelopes of the WORLD vocoder [75]. The spectral envelopes were extracted from the 16 kHz raw speech waveforms from the LibriSpeech corpus [78], varied from the 128, 256, 512, and 1024 bits/spectral envelope frame. The quantization performance was evaluated by Log Spectral Distortion (LSD). The Perceptual (PESQ) standardized as ITU-T recommendation [87] is also used to measure the quality of the reconstructed 16 kHz speech waveforms of the WORLD vocoder with or without spectral envelope quantization techniques. In the experimental results, the proposed GAN technique utilized to compare unquantized z-latent and quantized z-latent for embedding space updating can approximate the embedding space better than

the Mean Square Error of conventional VQ. The proposed model increases the average PESQ value by about 0.17 with a reduced average LSD of 0.5 dB with significant results compared to the VQ-VAE.

6.2 Methodology

6.2.1 Vector Quantized Variational AutoEncoder (VQ-VAE)

The Vector Quantized-Variational AutoEncoder (VQ-VAE) [44, 45] has been proposed as a VQ method based on deep learning, inspired by the conventional vector quantization, cooperated with the AutoEncoder (AE). The encoder network is constructed with the Convolutional Neural Network (CNN) to project the input data into the three-dimensional z-latent. The z-latent is reshaped into sub-vectors and applied with the VQ technique to quantize with the designed embedding space. The quantized, reshaped z-latent is then reshaped back to the original shape to represent the quantized z-latent. The decoder network transposes Convolutional Neural Network (transposed CNN) as the encoder network's counterpart. The quantized z-latent is the input of the decoder network to produce the output data.

The loss function is defined to update network parameters consisting of three terms. The VQ-VAE loss function is shown in Equation (6.1). The first term is the negative log-likelihood to optimize the encoder and decoder networks. The second term is the vector quantization error to optimize the embedding space. The commitment loss is the third term to force the encoder network to learn at the same speed as the embedding space.

$$L_{vqvae} = -\log\left(x\middle|z_q(x)\right) + \|sg[z_e(x)] - e\|_2^2 + \beta\|z_e(x) - sg[e]\|_2^2, \tag{6.1}$$

where x is input data, $z_q(x)$ is the quantized z-latent, $z_e(x)$ is the z-latent output of the encoder network, e is embedding space, β is a particular parameter that is set to 0.25, and $sg[.]$ is the stop gradient operator to freeze the parameter to be a constant and not changed in the backpropagation process.

6.2.2 Generative Adversarial Networks (GAN)

The Generative Adversarial Networks (GAN) [68] is an unsupervised deep learning technique to generate high-quality output. The GAN consists of the generator network to produce the generated data from random noise and the discriminator network to distinguish the real and generated data from the

generator network. The generator network tries to generate data similar to the real data, and the discriminator networks distinguish between real and generated data. This process is a game of two networks, and they try against each other by adapting the generator loss shown in Equation (6.2) to update parameters in the generator network and the discriminator loss in Equation (6.3) for the discriminator network.

$$\min_{G} L_{GAN}(G) = \mathrm{E}_{z\sim p_z} \log\left(1 - D(G(z))\right), \tag{6.2}$$

$$\max_{D} L_{GAN}(D) = \mathrm{E}_{x\sim p_d} \log\big(D(x)\big) + \mathrm{E}_{z\sim p_z} \log\left(1 - D(G(z))\right), \tag{6.3}$$

where $G(.)$ is the generator network, $D(.)$ is the discriminator network, z is random noise, P_z is the data distribution of random noise, x is the real data, and P_d is the distribution of the real data.

6.2.3 Deep learning parameter optimization

The Stochastic Gradient Decent (SGD) method [94, 95] is an ordinary optimizer to train deep learning parameters. However, when the number of the network parameters increases, the global minimum is hard to find, and the convergence speed becomes too slow. The Adam optimization [98, 99] is one of the solutions. The idea of gradient descent and momentum algorithm [96] and the Root Mean Square Propagation (RMSprop) [97] are combined to solve the convergence speed and better find the global minimum.

6.2.3.1 The gradient descent with the momentum algorithm

The gradient descent with the momentum algorithm [96] is based on a technique of exponentially weighted average and makes the convergence speed to the global minimum faster than the traditional SGD. Equation 6.4 shows deep learning weight parameter learning for the gradient descent with the momentum algorithm.

$$w_{t+1} = w_t - \alpha_t m_t \; ; \; m_t = \beta m_{t-1} + (1-\beta) \times \left[\frac{\partial L}{\partial w_t}\right], \tag{6.4}$$

where, m_t is the current time momentum of gradients, m_{t-1} is the previous time momentum of gradients, w_t is the current time weights, w_{t+1} is the next

time weights, α_t is the current time learning rate, $\frac{\partial L}{\partial w_t}$ is the derivative of the loss function with the weights at the current time, and the β is the moving average parameter.

6.2.3.2 Root Mean Square Propagation

The Root Mean Square propagation (RMSprop) optimizer [97] is based on a technique of exponential moving average over gradients. It is a very robust optimizer to find the global minimum, and the convergence speed is faster than the gradient descent with the momentum algorithm. Equation 6.5 shows the deep learning weight parameter learning for the RMSprop algorithm.

$$w_{t+1} = w_t - \frac{\alpha_t}{(v_t+\varepsilon)^{1/2}} \times \left[\frac{\partial L}{\partial w_t}\right] ; v_t = \beta v_{t-1} + (1-\beta) \times \left[\frac{\partial L}{\partial w_t}\right]^2, \quad (6.5)$$

where, w_t, w_{t+1} are the current and the next time weights, α_t is the current time learning rate, $\frac{\partial L}{\partial w_t}$ is the derivative of the loss function with the weights at the current time, v_t, v_{t-1} are the moving average of squared gradients at the current and the past time, β is the moving average parameter, and ε is constant.

6.2.3.3 Adam optimization

The Adam optimization [98, 99] is a variation of optimizers. The idea of gradient descent with momentum algorithm and the RMSprop are combined to improve the convergence speed and better find the global minimum. Equations 6.6 and 6.7 show m_t of the gradient descent with the momentum algorithm and v_t of RMSprop for Adam optimization. Equation 6.8 shows the deep learning weight parameter learning for the Adam algorithm by computing bias corrected $\widehat{m_t}$ and $\widehat{v_t}$.

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) \times \left[\frac{\partial L}{\partial w_t}\right], \quad (6.6)$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) \times \left[\frac{\partial L}{\partial w_t}\right]^2, \quad (6.7)$$

$$w_{t+1} = w_t - \widehat{m_t} \times \left(\frac{\alpha_t}{\sqrt{\widehat{v_t}+\varepsilon}}\right) ; \widehat{m_t} = \frac{m_t}{1-\beta_1} and\ \widehat{v_t} = \frac{v_t}{1-\beta_2}, \quad (6.8)$$

where, α_t is the current time learning rate, ε is constant, m_t, m_{t-1} are the current and previous time momentum of gradients, v_t, v_{t-1} are the moving average of squared gradients at the current and the past time, $\frac{\partial L}{\partial w_t}$ is the derivative of loss function with the weights at the current time, and β_1 and β_2 are decay rates.

6.2.4 Variational AutoEncoder Generative Adversarial Networks (VAEGAN)

The combination of the VAE with GA has been studied in [48], which is called the Variational AutoEncoder Generative Adversarial Networks (VAEGAN). The model consists of the encoder network, the decoder network, and the discriminator network. The encoder network transforms the input data into z-latents. The random noise is a normal distribution, constructed as the same dimension as the z-latent to compute the KL divergence loss, and the encoder network produces the z-latent as random noise. The decoder network adopts the z-latent as the input and reconstructs the output related to the input data. As the results in [18], a standard AE produces a poor quality of the reconstructed output and blurred images. The discriminator of the GAN technique operates as a reconstruction performance-enhancing method. The discriminator obtains the input data and reconstructs output data to improve reconstruction performance. It controls the adversarial loss to update the decoder network parameters by backpropagation.

6.2.5 The VAEGAN implemented in the Vector Quantized Variational AutoEncoder (VAEGAN implemented in VQ-VAE)

Figure 6.1 shows the procedure of the proposed VAEGAN implemented in the Vector Quantized Variational AutoEncoder (VAEGAN implemented in VQ-VAE). The input x_t (SP vector) is fed into the encoder network to produce the unquantized z-latent $z_e(x_t)$. It is reshaped and quantized with the embedding space e by choosing the nearest vector pattern. The index in the codebook of the embedding space e is transmitted to the decoder. The decoder reproduces the quantized z-latent $z_q(x_t)$ and generates the output $\tilde{x}_t$ (Quantized SP vector). The difference from the other techniques is that the VAEGAN implemented in the Vector Quantized Variational AutoEncoder draws the sample from a normal distribution z_{p_t} with the same shape with $z_e(x_t)$ and $z_q(x_t)$, then feed into the decoder network to produce the $\tilde{x}_{p_t}$ for calculating the KL divergence loss in Equation 6.11:

$$L_{KL} = z_e(x_t) \times \left(\log\left(z_e(x_t)\right) - \log\left(z_{p_t}\right)\right). \quad (6.11)$$

Algorithm The VAEGAN implemented in VQ-VAE. All experiment in the paper used the default values $\alpha = 0.0001$, $m = 32$, $K = 256$, $\gamma = 10$. $\beta = 0.25$.

Require: α, the learning rate. m, the batch size. E_θ, the encoder network. $D_\emptyset$, the decoder network. e_φ, the embedding space. Dis_ρ, the discriminator network. K, number of vector patterns. β, weight parameter. γ, VAEGAN weight parameter.

Initialize E_θ with random weight θ

$D_\emptyset$ with random weight $\emptyset$

Dis_ρ with random weight ρ

e_φ with random weight φ

for t = 1, 2, …, N **do**

x_t = Sample $\{x^{(i)}\}_{i=1}^{m}$ ~ random a batch of input data

$z_e(x_t) = E_\theta(x_t)$

$z_{p_t} = samples\ from\ normal\ distribution$

$z_{e_{reshaped}}(x_t) = reshaping\big(z_e(x_t)\big)$

$idx_t = arg\min_k \left\| z_{e_{reshaped}}(x_t) - e_{\varphi_k} \right\|_2^2$

$e_{\varphi_{idx_t}} = idx_t$

$z_{e_{reshaped}}(x_t)_{quantized} = e_{\varphi_{idx_t}}$

$z_q(x_t) = inv.reshaping(z_{e_{reshaped}}(x_t)_{quantized})$

$\hat{x}_t = D_\emptyset(z_q(x_t))$

$\hat{x}_{p_t} = D_\emptyset(z_{p_t})$

$L_{KL} = D_{KL}(Q(z_{p_t}|x_t)||P(z_e(x_t)|x_t))$

$$L = -\log p(x_t|z_q(x_t)) + \left\| sg\left[z_{e_{reshaped}}(x_t)\right] - e_{\varphi_{idx_t}} \right\|_2^2 + \beta \left\| z_{e_{reshaped}}(x_t) - sg[e_{\varphi_{idx_t}}] \right\|_2^2$$

$$L_{GAN} = \log(Dis_\rho(x_t)) + \log(1 - Dis_\rho(\hat{x}_t)) + \log(1 - Dis_\rho(\hat{x}_{p_t}))$$

$L_{en+em} = L_{KL} + L$

$L_{de} = (\gamma L) - L_{GAN}$

$L_{dis} = L_{GAN}$

$\theta_{t+1}, \varphi_{t+1} \leftarrow Adam(\alpha, L_{en+em}, \theta_t, \varphi_t)$

$\emptyset_{t+1} \leftarrow Adam(\alpha, L_{de}, \emptyset_t)$

$\rho_{t+1} \leftarrow Adam(\alpha, L_{dis}, \rho_t)$

End for

Figure 6.1: The procedure of VAEGAN implemented in VQ-VAE algorithm.

The second loss term L in Equation 6.1 is the calculation of the loss by the Adam optimizer in Equation 6.8. The first loss term L is computed from the input x_t (SP vector), the output $\tilde{x}_t$ (Quantized SP vector), unquantized z-latent $z_e(x_t)$ and quantized z-latent $z_q(x_t)$. The discriminator network organizes to discriminate between the unquantized SP vector x_t and the quantized SP vector $\tilde{x}_t$. The adversarial loss can calculate from Equation 6.12.

$$L_{GAN} = \log\big(Dis(x_t)\big) + \log\big(1 - Dis(\tilde{x}_t)\big) + \log\Big(1 - Dis\big(\tilde{x}_{p_t}\big)\Big), \quad (6.12)$$

where log(.) is the logarithmic function. $Dis(.)$ is the discriminator network. x_t is the unquantized SP vector at time t. $\tilde{x}_t$ is the quantized SP vector. The $\tilde{x}_{p_t}$ is the output from the decoder network that utilizes random noise z_p as input.

Equation 6.13 shows the loss term to update the encoder network parameters and embedding space parameters by Adam optimizer in Equation 6.8 presented in Equation 6.13. The decoder network parameters are updated by Adam optimizer in Equation 6.8 by using the loss term in Equation 6.14, where γ is the weight set to 10. The discriminator network parameters are updated by Equation 6.15 based on Adam optimizer Equation 6.8.

$$L_{en+em} = L_{KL} + L, \quad (6.13)$$

$$L_{de} = (\gamma L) - L_{GAN}, \quad (6.14)$$

$$L_{dis} = L_{GAN}, \quad (6.15)$$

In the training process, the mini-batch with 32 samples passes through the VAEGAN implemented in VQ-VAE network parameters in every iteration. The loss terms update the network parameters by Adam optimizer.

The loss term L_{en+em} in Equation 6.13 is utilized to update the encoder network parameters by the Adam optimizer presented in Equations 6.16, 6.17, and 6.18 for computing the momentum of gradients m_{θ_t}, the moving average of squared gradients v_{θ_t}, and the next estimated encoder network weight parameters θ_{t+1}, respectively.

$$m_{\theta_t} = \beta_1 m_{\theta_{t-1}} + (1 - \beta_1) \times \left[\frac{\partial L_{en+em}}{\partial \theta_t}\right],$$

(6.16)

$$v_{\theta_t} = \beta_2 v_{\theta_{t-1}} + (1-\beta_2) \times \left[\frac{\partial L_{en+em}}{\partial \theta_t}\right]^2, \tag{6.17}$$

$$\theta_{t+1} = \theta_t - \widehat{m_{\theta_t}} \times \left(\frac{\alpha_t}{\sqrt{\widehat{v_{\theta_t}}+\varepsilon}}\right); \widehat{m_{\theta_t}} = \frac{m_{\theta_t}}{1-\beta_1} and \ \widehat{v_{\theta_t}} = \frac{v_{\theta_t}}{1-\beta_2}, \tag{6.18}$$

where, α_t is the current time learning rate, ε is constant, m_{θ_t}, $m_{\theta_{t-1}}$ are the current and the previous time momentum of gradients of the encoder network, $v_{\theta_t}, v_{\theta_{t-1}}$ are the current and the past moving averages of squared gradients of the encoder network, $\frac{\partial L_{L_{en+em}}}{\partial \theta_t}$ is the derivative of the encoder network and embedding space loss function with the weights at the current time, and the β_1 and β_2 are decay rates.

The loss term L_{en+em} in Equation 6.13 also is applied to update the embedding space parameters by Adam optimizer presented in Equations 6.19, 6.20, and 6.21 for computing the momentum of gradients m_{φ_t}, the moving average of squared gradients v_{φ_t}, and the next estimated embedding space weight parameters φ_{t+1}, respectively.

$$m_{\varphi_t} = \beta_1 m_{\varphi_{t-1}} + (1-\beta_1) \times \left[\frac{\partial L_{en+em}}{\partial \varphi_t}\right], \tag{6.19}$$

$$v_{\varphi_t} = \beta_2 v_{\varphi_{t-1}} + (1-\beta_2) \times \left[\frac{\partial L_{en+em}}{\partial \varphi_t}\right]^2, \tag{6.20}$$

$$\varphi_{t+1} = \varphi_t - \widehat{m_{\varphi_t}} \times \left(\frac{\alpha_t}{\sqrt{\widehat{v_{\varphi_t}}+\varepsilon}}\right); \widehat{m_{\varphi_t}} = \frac{m_{\varphi_t}}{1-\beta_1} \ and \ \widehat{v_{\varphi_t}} = \frac{v_{\varphi_t}}{1-\beta_2}, \tag{6.21}$$

where, α_t is the current time learning rate, ε is constant, m_{φ_t}, $m_{\varphi_{t-1}}$ are the current and the past time momentum of gradients of embedding space, v_{φ_t}, $v_{\varphi_{t-1}}$ are the current and the past moving averages of squared gradients of the embedding space, $\frac{\partial L_{L_{en+em}}}{\partial \varphi_t}$ is the derivative of the encoder network and embedding space loss functions with the weights at the current time, and the β_1 and β_2 are decay rates.

The loss term L_{de} in Equation 6.14 is utilized to update the decoder network parameters by Adam optimizer presented in Equations 6.22, 6.23,

6.24 for computing the momentum of gradients $m_{\emptyset_t}$, the moving average of squared gradients $v_{\emptyset_t}$, and the next guess decoder network weight parameters $\emptyset_{t+1}$, respectively.

$$m_{\emptyset_t} = \beta_1 m_{\emptyset_{t-1}} + (1-\beta_1) \times \left[\frac{\partial L_{de}}{\partial \emptyset_t}\right], \tag{6.22}$$

$$v_{\emptyset_t} = \beta_2 v_{\emptyset_{t-1}} + (1-\beta_2) \times \left[\frac{\partial L_{de}}{\partial \emptyset_t}\right]^2, \tag{6.23}$$

$$\emptyset_{t+1} = \emptyset_t - \widehat{m_{\emptyset_t}} \times \left(\frac{\alpha_t}{\sqrt{\widehat{v_{\emptyset_t}}+\varepsilon}}\right); \widehat{m_{\emptyset_t}} = \frac{m_{\emptyset_t}}{1-\beta_1} \; and \; \widehat{v_{\emptyset_t}} = \frac{v_{\emptyset_t}}{1-\beta_2}, \tag{6.24}$$

where, α_t is the current time learning rate, the ε is constant, the $m_{\emptyset_t}, m_{\emptyset_{t-1}}$ are the current and previous time momentum of gradients of the decoder network, $v_{\emptyset_t}, v_{\emptyset_{t-1}}$ are the current and the past time moving averages of squared gradients of the decoder network, $\frac{\partial L_{de}}{\partial \emptyset_t}$ is the derivative of the decoder network loss function with the weights at the current time, and the $\beta_1$and β_2 are decay rates.

The loss term L_{dis} in Equation 6.15 is utilized to update the discriminator network parameters by Adam optimizer presented in Equations 6.25, 6.26, 6.27 for computing the momentum of gradients m_{ρ_t}, the moving average of squared gradients v_{ρ_t}, and the next estimated discriminator network weight parameters ρ_{t+1}, respectively.

$$m_{\rho_t} = \beta_1 m_{\rho_{t-1}} + (1-\beta_1) \times \left[\frac{\partial L_{dis}}{\partial \rho_t}\right], \tag{6.25}$$

$$v_{\rho_t} = \beta_2 \, v_{\rho_{t-1}} + (1-\beta_2) \times \left[\frac{\partial L_{dis}}{\partial \rho_t}\right]^2, \tag{6.26}$$

$$\rho_{t+1} = \rho_t - \widehat{m_{\rho_t}} \times \left(\frac{\alpha_t}{\sqrt{\widehat{v_{\rho_t}}+\varepsilon}}\right); \widehat{m_{\rho_t}} = \frac{m_{\rho_t}}{1-\beta_1} \; and \; \widehat{v_{\rho_t}} = \frac{v_{\rho_t}}{1-\beta_2}, \tag{6.27}$$

where, α_t is the current time learning rate, ε is constant, $m_{\rho_t}, m_{\rho_{t-1}}$ are the current and previous time momentum of gradients of the discriminator network, v_{ρ_t}, $v_{\rho_{t-1}}$ are the current and the past time moving average of squared gradients of the discriminator network, $\frac{\partial L_{dis}}{\partial \rho_t}$ is the derivative of discriminator network loss function with the weights at the current time, and $\beta_1$and $\beta_2$are

decay rates.

6.2.6 The Vector Quantized Variational AutoEncoder with Embedding space Learned by Generative Adversarial Networks (VQ-VAE-EMGAN)

The conventional VQ-VAE consists of an encoder network, a decoder network, and an embedding space. The encoder network receives the input data to produce the z-latent and reshapes the z-latent into sub-vectors fitted to the vector quantization task in embedding space. The vector quantization organizes the discrete z-latent. The quantized z-latent is constructed based on the discrete version z-latent to pick the corresponding vector patterns in the embedding space. The z-latent is used as the decoder network's input to reconstruct the output data. The loss function criteria of three terms in Equation 6.1 are applied to update network parameters. The second term, called VQ loss, is applied to update embedding space parameters based on the mean square error between unquantized z-latent and quantized z-latent. The quality of quantized z-latent depends on the minimization of mean square errors. The more similarity between unquantized z-latent and quantized z-latent provides a better reconstruction performance of the output data. The adversarial loss of the GAN technique is the progressive of deep learning loss criteria. The advantage of using the adversarial loss is that the model can generate realistic data or high-definition data compared to the AE-generated blurry data [73].

We propose the Vector Quantized Variational AutoEncoder with Embedding space learned by Generative Adversarial Networks (VQ-VAE-EMGAN) to replace the VQ loss based on the mean square errors with the adversarial loss based on the GAN technique.

Figure 6.2 shows the procedure of the VQ-VAE-EMGAN algorithm. The input x_t (SP vector) is fed into the encoder network to produce the unquantized z-latent $z_e(x_t)$. It is reshaped and quantized with the embedding space e. The index in the codebook of the embedding space e is transmitted, and the decoder reproduces the quantized z-latent $z_q(x_t)$ by choosing the corresponded vector pattern in the embedding space e as the input to the decoder network to reproduce the output $\tilde{x}_t$ (Quantized SP vector). The loss L_{vqvae} in Equation 6.1 is applied to calculate the loss to update the encoder network parameters and decoder network parameters with the Adam optimizer in Equation 6.8. The first loss term L (at line 11, Figure 6.2) is computed from the input x_t (SP vector), the output $\tilde{x}_t$ (Quantized SP vector), unquantized z-

latent $z_e(x_t)$ and quantized z-latent $z_q(x_t)$. The discriminator network organizes to discriminate between the unquantized z-latent $z_e(x_t)$ and the quantized z-latent $z_q(x_t)$. The generator loss L_G (at line 12, Figure 6.2) is computed by Equation 6.28, and the discriminator loss L_D (at line 13, Figure 6.2) is computed by Equation 6.29:

$$L_G = \log(1 - Dis(z_q(x_t))), \tag{6.28}$$

$$L_D = \log(z_e(x_t)) + \log(1 - Dis(z_q(x_t))), \tag{6.29}$$

where $\log(.)$ is the logarithmic function. The $Dis(.)$ is the discriminator network. The $z_e(x_t)$ is the unquantized z-latent. The $z_q(x_t)$ is the quantized z-latent.

The adversarial loss L_G from the discriminator is calculated and the Adam optimizer in Equation 6.8 updates the embedding space parameters e in the generator network. The adversarial loss L_D from the discriminator is applied to update discriminator network parameters by Adam optimizer in Equation 6.8.

The mini-batch with 32 samples passed through the VQ-VAE-EMGAN network parameters in the training process in every iteration. The loss terms are updated the network parameters by using an Adam optimizer similar to the VAEGAN implemented in VQ-VAE but follows the procedure in Figure 6.2 instead.

Algorithm The VQ-VAE-EMGAN. All experiment in the paper used the default values $\alpha = 0.0001$, $m = 32$, $K = 256$, $\beta = 0.25$.

Require: α, the learning rate. m, the batch size. E_θ, the encoder network. $D_\emptyset$, the decoder network. e_φ, the embedding space. Dis_ρ, the discriminator network. K, number of vector patterns. β, weight parameter.

Initialize E_θ with random weight θ

$D_\emptyset$ with random weight $\emptyset$

Dis_ρ with random weight ρ

e_φ with random weight φ

for t = 1, 2, …, N **do**

x_t = Sample $\{x^{(i)}\}_{i=1}^{m}$ ~ random a batch of input data

$z_e(x_t) = E_\theta(x_t)$

$z_{e_{reshaped}}(x_t) = reshaping\left(z_e(x_t)\right)$

$idx_t = arg\min_k \left\| z_{e_{reshaped}}(x_t) - e_{\varphi_k} \right\|_2^2$

$e_{\varphi_{idx_t}} = idx_t$

$z_{e_{reshaped}}(x_t)_{quantized} = e_{\varphi_{idx_t}}$

$z_q(x_t) = inv.reshaping(z_{e_{reshaped}}(x_t)_{quantized})$

$\tilde{x}_t = D_\emptyset(z_q(x_t))$

$L = -\log p(x_t|z_q(x_t)) + \left\| sg\left[z_{e_{reshaped}}(x_t)\right] - e_{\varphi_{idx_t}} \right\|_2^2 + \beta \left\| z_{e_{reshaped}}(x_t) - sg[e_{\varphi_{idx_t}}] \right\|_2^2$

$L_G = \log(1 - Dis_\rho(z_q(x_t)))$

$L_D = \log(Dis_\rho(z_e(x_t))) + \log(1 - Dis_\rho(z_q(x_t)))$

$\theta_{t+1}, \emptyset_{t+1} \leftarrow Adam(\alpha, L, \theta_t, \emptyset_t)$

$\varphi_{t+1} \leftarrow Adam(\alpha, L_G, \varphi_t)$

$\rho_{t+1} \leftarrow Adam(\alpha, L_D, \rho_t)$

End for

Figure 6.2: The procedure of the VQ-VAE-EMGAN algorithm.

6.2.7 The Vector Quantized Variational AutoEncoder with Embedding space and Decoder Network Learned by Generative Adversarial Networks (VQ-VAE-EMDEC)

Figure 6.3 shows the procedure of the proposed Vector Quantized Variational AutoEncoder with Embedding space and Decoder Network Learned by Generative Adversarial Networks (VQ-VAE-EMDEC). The input x_t (SP vector) is fed into the encoder network to produce the unquantized z-latent $z_e(x_t)$. It is reshaped and quantized with the embedding space e. The index in the codebook of the embedding space e is transmitted and reproduce the quantized z-latent $z_q(x_t)$ by choosing the corresponding vector pattern in the embedding space e as the input to the decoder network to reproduce the output $\tilde{x}_t$ (Quantized SP vector). The VQ-VAE-EMDEC technique is the same as the VAEGAN implemented in VQ-VAE. It draws the sample from the normal distribution z_{p_t} with the same shape with $z_e(x_t)$ and $z_q(x_t)$, then the z-latent is fed into the decoder network to produce the $\tilde{x}_{p_t}$ for calculating the KL divergence loss in Equation 6.30:

$$L_{KL} = z_e(x_t) \times \big(\log\big(z_e(x_t)\big) - \log\big(z_{p_t}\big)\big). \tag{6.30}$$

The second loss term L (at line 14, Figure 6.3) is the form of Equation 6.1, and it is calculated with the Adam optimizer in Equation 6.8. The loss term L is computed from the input x_t (SP vector), the output $\tilde{x}_t$ (Quantized SP vector), unquantized z-latent $z_e(x_t)$ and quantized z-latent $z_q(x_t)$. The discriminator network is organized to discriminate between the unquantized SP vector x_t and the quantized SP vector $\tilde{x}_t$. The adversarial loss is defined as Equation 6.31:

$$L_{GAN} = \log\big(Dis(x_t)\big) + \log\big(1 - Dis(\tilde{x}_t)\big) + \log\Big(1 - Dis\big(\tilde{x}_{p_t}\big)\Big), \tag{6.31}$$

where log(.) is the logarithmic function. $Dis(.)$ is the discriminator network. x_t is the unquantized SP vector. $\tilde{x}_t$ is the quantized SP vector. $\tilde{x}_{p_t}$ is the output from the decoder network by using random noise z_{p_t} as input.

The loss terms L_{en} (at line 16, Figure 6.3) are used to update the encoder network parameters by Adam optimizer presented in Equation 6.32. L_{de+em} (at line 17, Figure 6.3) is the loss to update the decoder network parameters and embedding space parameters as in Equation 6.33, where γ is a weight set to 10. L_{dis} (at line 18, Figure 6.3) is the loss to update the discriminator

network parameters by Equation 6.34.

$$L_{en} = L_{KL} + L, \tag{6.32}$$

$$L_{de+em} = (\gamma L) - L_{GAN}, \tag{6.33}$$

$$L_{dis} = L_{GAN}, \tag{6.34}$$

In the training process, the mini-batch with 32 samples passed through the VQ-VAE-EMDEC network parameters in every iteration. The loss terms updated the network parameter by similar Adam optimizer to the VAEGAN implemented in VQ-VAE but followed the procedure in Figure 6.3 instead.

Algorithm The VQ-VAE-EMDEC. All experiment in the paper used the default values $\alpha = 0.0001, m = 32, K = 256, \gamma = 10. \beta = 0.25.$

Require: α, the learning rate. m, the batch size. E_θ, the encoder network. $D_\emptyset$, the decoder network. e_φ, the embedding space. Dis_ρ, the discriminator network. K, number of vector patterns. β, weight parameter. γ, VAEGAN weight parameter.

Initialize E_θ with random weight θ

$D_\emptyset$ with random weight $\emptyset$

Dis_ρ with random weight ρ

e_φ with random weight φ

for t = 1, 2, …, N **do**

x_t = Sample $\{x^{(i)}\}_{i=1}^{m} \sim$ random a batch of input data

$z_e(x_t) = E_\theta(x_t)$

$z_{p_t} = samples\ from\ normal\ distribution$

$z_{e_{reshaped}}(x_t) = reshaping\big(z_e(x_t)\big)$

$idx_t = arg\min_k \left\| z_{e_{reshaped}}(x_t) - e_{\varphi_k} \right\|_2^2$

$e_{\varphi_{idx_t}} = idx_t$

$z_{e_{reshaped}}(x_t)_{quantized} = e_{\varphi_{idx_t}}$

$z_q(x_t) = inv.reshaping(z_{e_{reshaped}}(x_t)_{quantized})$

$\hat{x}_t = D_\emptyset(z_q(x_t))$

$\hat{x}_{p_t} = D_\emptyset(z_{p_t})$

$L_{KL} = D_{KL}(Q(z_{p_t}|x_t)||P(z_e(x_t)|x_t))$

$$L = -\log p(x_t|z_q(x_t)) + \left\| sg\left[z_{e_{reshaped}}(x_t)\right] - e_{\varphi_{idx_t}} \right\|_2^2 + \beta \left\| z_{e_{reshaped}}(x_t) - sg[e_{\varphi_{idx_t}}] \right\|_2^2$$

$$L_{GAN} = \log(Dis_\rho(x_t)) + \log(1 - Dis_\rho(\hat{x}_t)) + \log(1 - Dis_\rho(\hat{x}_{p_t}))$$

$L_{en} = L_{KL} + L$

$L_{de+em} = (\gamma L) - L_{GAN}$

$L_{dis} = L_{GAN}$

$\theta_{t+1} \leftarrow Adam(\alpha, L_{en}, \theta_t)$

$\emptyset_{t+1}, \varphi_{t+1} \leftarrow Adam(\alpha, L_{de+em}, \emptyset_t, \varphi_t)$

$\rho_{t+1} \leftarrow Adam(\alpha, L_{dis}, \rho_t)$

End for

Figure 6.3: The procedure of VQ-VAE-EMDEC algorithm.

6.3 Experiments and Results

The experiment on the quantizer design was conducted. The four methods were compared: (1) the VQ-VAE (reference), (2) the VAEGAN implemented in VQ-VAE, (3) the VQ-VAE-EMGAN, and (4) the VQ-VAE-EMDEC.

The (1) VQ-VAE is the reference method in which no GAN techniques are included. The proposed methods (2), (3), and (4) are trained by deep learning with various adversarial techniques. The main structures of the encoder, the quantization codebook, and the decoder are unchanged for the four methods. Since the GAN changes loss functions, all the parameters are trained, respectively.

The data for this experiment was the WORLD vocoder's spectral envelope parameters from 16 kHz sampling speech waveforms. Each model of them has four models with different target bitrates.

6.3.1 The raw speech waveform database

The experiments were conducted with the LibriSpeech ASR corpus [78] of 16 kHz sampling English speech. The clean speech in the development set was selected as the training set. The dev-clean database consists of 8.97 hours with 40 speakers (2,719 waveforms), and the test-clean one, 8.56 hours with 39 speakers (2,620 waveforms). In the model training process, we utilized the full dev-clean dataset, and in the evaluation process, we randomly selected 100 waveforms from the test-clean dataset to evaluate the quantization performance. The WORLD vocoder [75] extracted the spectral envelope parameters from every five milliseconds of raw speech waveforms, and preprocesses were applied for training models.

6.3.2 The spectral envelope parameter quantization

Figure 6.4 shows the procedure of spectral envelope parameter quantization. The WORLD vocoder framework was applied to evaluate spectral envelope quantization performance of 16 kHz raw speech waveforms. The speech analysis part extracted speech parameters from every five milliseconds of speech waveforms, the constant value of fundamental frequency (F0) with the shape of height, width, depth as (H=1, W=1, D=1), the vector of spectral envelopes parameter (SP) as (H=1, W=513, D=1), and the vector of the aperiodic parameter (AP) as (H=1, W=513, D=1). The output of 5 milliseconds was reconstructed from the speech synthesis with those speech

parameters. In this paper, since we focused on the performance of the SP parameter quantization, the F0 and AP parameters were unquantized and directly used in synthesizing speech at the decoder. For the SP parameters, the logarithmic base ten and the min-max normalization were applied to normalize the scaling of the values between 0 to 1. The normalized SP parameter was defined as SP_{norm} in Equation 6.35.

$$SP_{norm} = \frac{log_{10}(SP) - Min_{SP}}{Max_{SP} - Min_{SP}}, \tag{6.35}$$

where SP is the original spectral envelope parameter, $\log_{10}(.)$ is the logarithmic base ten, Min_{SP} is the minimum value of SP, Max_{SP} is the maximum value of SP, and the SP_{norm} is the normalized spectral envelope parameter.

The quantized normalized SP parameter SP_{norm_q} was obtained from the VQ method and applied the inverse min-max normalization in Equation 6.36 followed by the inverse logarithmic base ten in Equation 6.37 to return the original values for representing the quantized SP parameter SP_q The speech parameters consisted of F0, SP_q and AP were the input of the synthesis process to reconstruct the five milliseconds output speech waveform.

$$SP_{\log(10)_q} = [SP_{norm_q} \times (Max_{SP} - Min_{SP})] + Min_{SP}, \tag{6.36}$$

$$SP_q = 10^{SP_{\log(10)_q}}, \tag{6.37}$$

where SP_{norm_q} is the quantized normalized SP parameter, Min_{SP} is the minimum value of SP, Max_{SP} is the maximum value of SP, $SP_{\log(10)_q}$ is the quantized logarithm base ten domain of SP parameter, and SP_q is the quantized SP parameter.

The VQ-VAE, the VAEGAN implemented in VQ-VAE, the VQ-VAE-EMGAN, and the VQ-VAE-EMDEC were designed to quantize the spectral envelope parameter of the WORLD vocoder with the fixed embedding space of 256 vector patterns (8 bits quantizer).

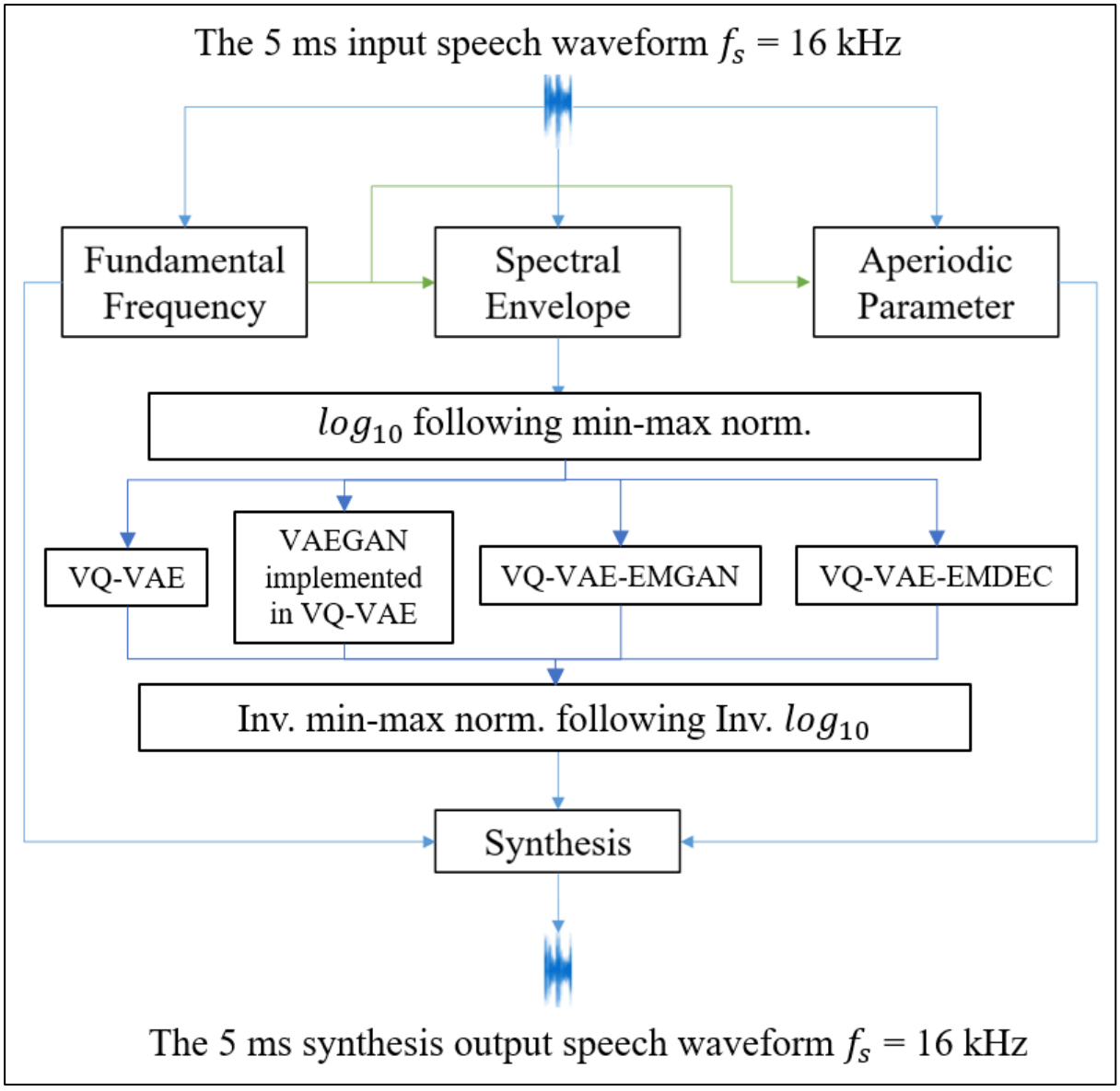

Figure 6.4: The block diagram of the WORLD spectral envelope parameter quantization.

Table 6.1 shows the common deep learning architecture of the encoder and decoder networks for the VQ-VAE, the VAEGAN implemented in VQ-VAE, the VQ-VAE-EMGAN, and the VQ-VAE-EMDEC methods. The encoder consists of two CNN input layers with stride 2, two residual networks, and the prepared VQ layer to force the output channel to be the same as the length of vector patterns in the embedding space. The decoder is the inverse version of the encoder network consisting of the CNN input layer connected with two residual networks and two transposed CNN layers. The decoder transforms the received z-latent data into the fitted size of the next convolutional layers to calculate the summation into a single constant.

Table 6.1
The encoder network and decoder network architectures of four implemented techniques.

Encoder network	
Layer	Architecture
Input_1	4×4 128 Conv, stride 2, ReLU
Input_2	4×4 256 Conv, stride 2, ReLU
Residual1_1	3×3 128 Conv, stride 1, ReLU
Residual1_2	1×1 256 Conv, stride 1, ReLU
Residual2_1	3×3 128 Conv, stride 1, ReLU
Residual2_2	1×1 256 Conv, stride 1, ReLU
Pre_VQ	1×1 32 Conv, stride 1
Decoder network	
Layer	Architecture
Input_1	3×3 256 Conv, stride 1, ReLU
Residual1_1	3×3 128 Conv, stride 1, ReLU
Residual1_2	1×1 256 Conv, stride 1, ReLU
Residual2_1	3×3 128 Conv, stride 1, ReLU
Residual2_2	1×1 256 Conv, stride 1, ReLU
Output_1	4×4 128 Conv, stride 2, ReLU
Output_2	4×4 1 Conv, stride 2, Sigmoid

Figure 6.5 shows the quantization process of the VQ-VAE. The SP vector was the input of the encoder network to produce the z-latent matrix. First, the reshaping method was applied to the z-latent matrix for the reshaped z-latent vectors with the vector length corresponding to the vector length of the designed embedding space. Then, the VQ was applied to transform the continuous reshaped z-latent vectors into a discrete presentation, and the inverse VQ was transformed back to the quantized z-latent matrix as quantized reshaped z-latent vectors. The input of the decoder network was the quantized z-latent matrix that was reshaped from the quantized reshaped z-latent vectors and reproduced the quantized SP vector.

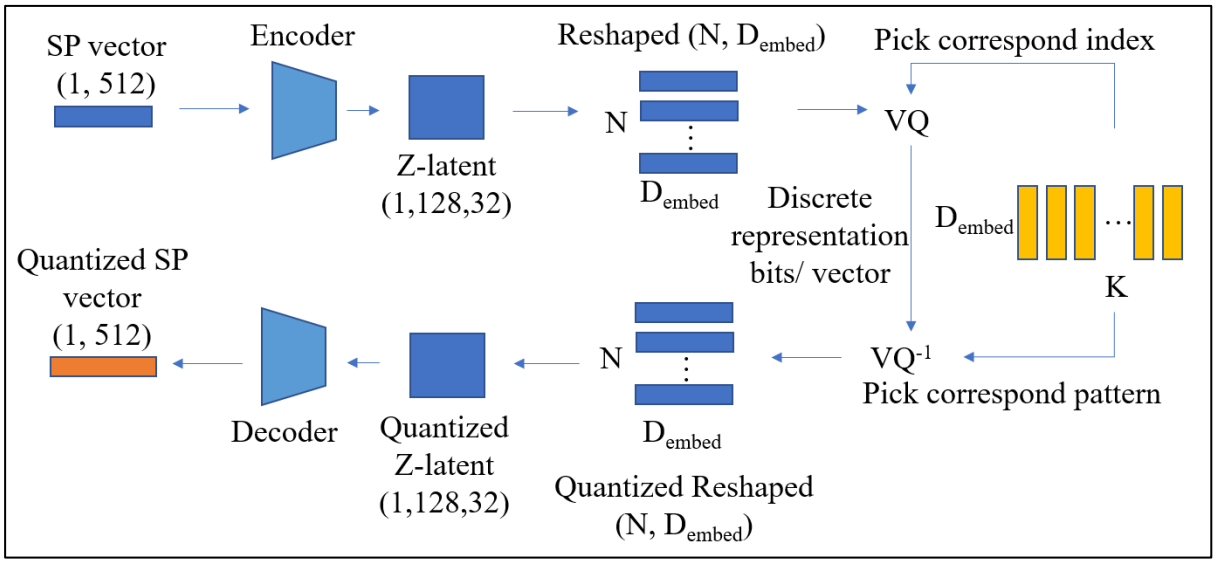

Figure 6.5: The VQ-VAE diagram.

Figure 6.6 shows the VQ-VAE training process. In training, the encoder network receives the input SP vector and produces the z-latent, which minimizes Euclidean distance, and the index is sent to the decoder. The decoder network receives the index and reconstructs the quantized z-latents by picking the corresponding vector pattern in the embedding space. In the end, Equation 6.1 calculates the VQ-VAE loss, and the Adam optimizer updates network parameters consisting of the encoder network, decoder network, and embedding space (codebook). The training of VQ-VAE models uses the learning rate parameter: 0.0001, the optimizer: Adam, the number of training epochs: 10, and minibatch: the random of thirty-two SP vectors from the SP vector database.

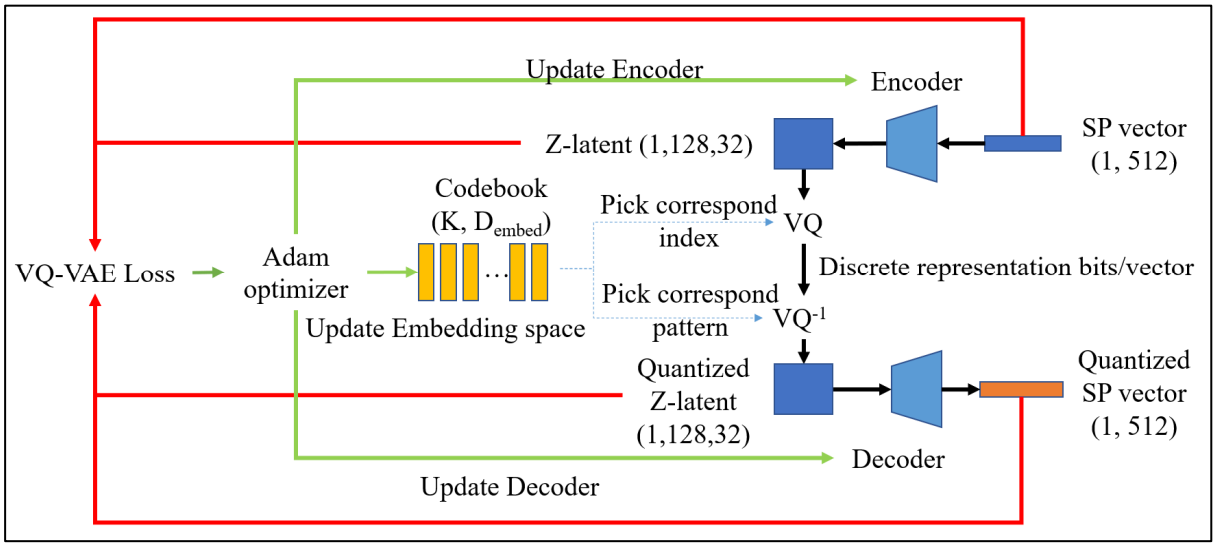

Figure 6.6: The VQ-VAE training process.

Figure 6.7 shows the quantization process of the VAEGAN implemented VQVAE, the VQ-VAE-EMGAN, and the VQ-VAE-EMDEC. The SP vector was the input of the encoder network to produce the z-latent matrix. The reshaping method was applied to the z-latent matrix for the reshaped z-latent vectors with the vector length corresponding to the vector length of the designed embedding space. Next, the VQ was applied to transform the continuous reshaped z-latent vectors into a discrete presentation, and the inverse VQ was transformed back to the quantized z-latent matrix as quantized reshaped z-latent vectors. The input of the decoder network was the quantized z-latent matrix that was reshaped from the quantized reshaped z-latent vectors and reproduced the quantized SP vector.

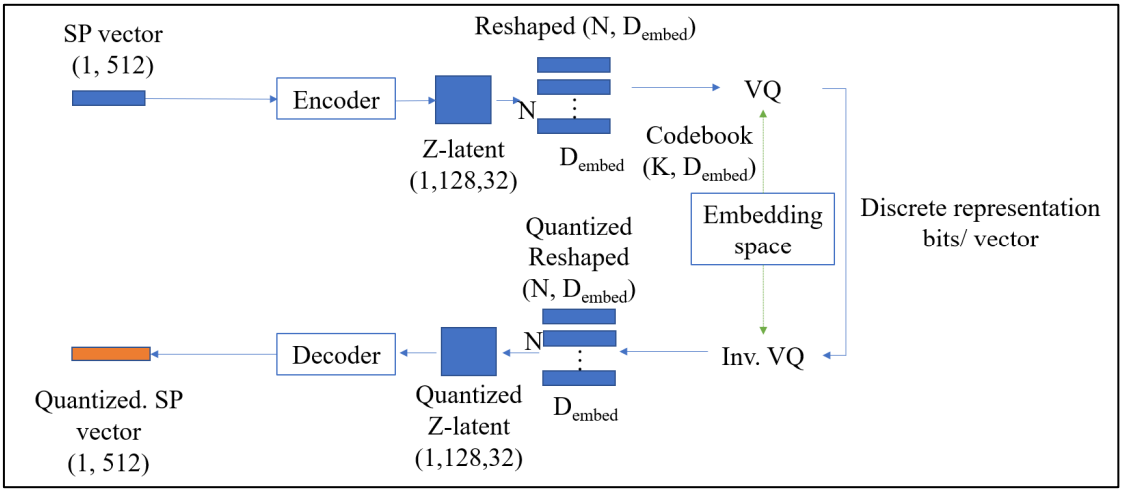

Figure 6.7: The quantization diagram of the VAEGAN implemented VQVAE, the VQ-VAE-EMGAN, and the VQ-VAE-EMDEC.

Figure 6.8 shows the training process of the VAEGAN implemented in VQ-VAE. The input SP vector is fed into the encoder network to produce the unquantized z-latent. It is reshaped and quantized with the embedding space. The index of the codebook of the embedding space is transmitted and reproduces the quantized z-latent by choosing the corresponding vector pattern in the embedding space as the input to the decoder network to reproduce the quantized SP vector). The difference from the other techniques is that the VAEGAN implemented in the VQ-VAE draws the sample from a normal distribution with the same shape as the z-latent and then feeds into the decoder network to produce the SP vector from the normal distribution for calculating the KL divergence loss in Equation 6.11. The second loss term L (at line 14, Figure 6.1), defined by Equation 6.1, calculates the loss with the Adam optimizer in Equation 6.8. The discriminator network is organized to discriminate between the unquantized and the quantized SP vectors. Equation 6.12 can calculate the adversarial loss. The loss term updates the encoder network parameters and embedding space parameters with the Adam optimizer in Equation 6.8 presented in Equation 6.13. The decoder network parameters are updated with the Adam optimizer Equation 6.8 from the loss term in Equation 6.14, where the γ is the weight set to 10. The discriminator network parameters are updated by Equation 6.15 with the Adam optimizer in Equation 6.8. The training of The VAEGAN implemented in VQ-VAE models uses the following parameters: the learning rate: 0.0001, the optimizer: Adam, the number of training epochs: 10, and minibatch: the random of thirty-two SP vectors from the SP vector database.

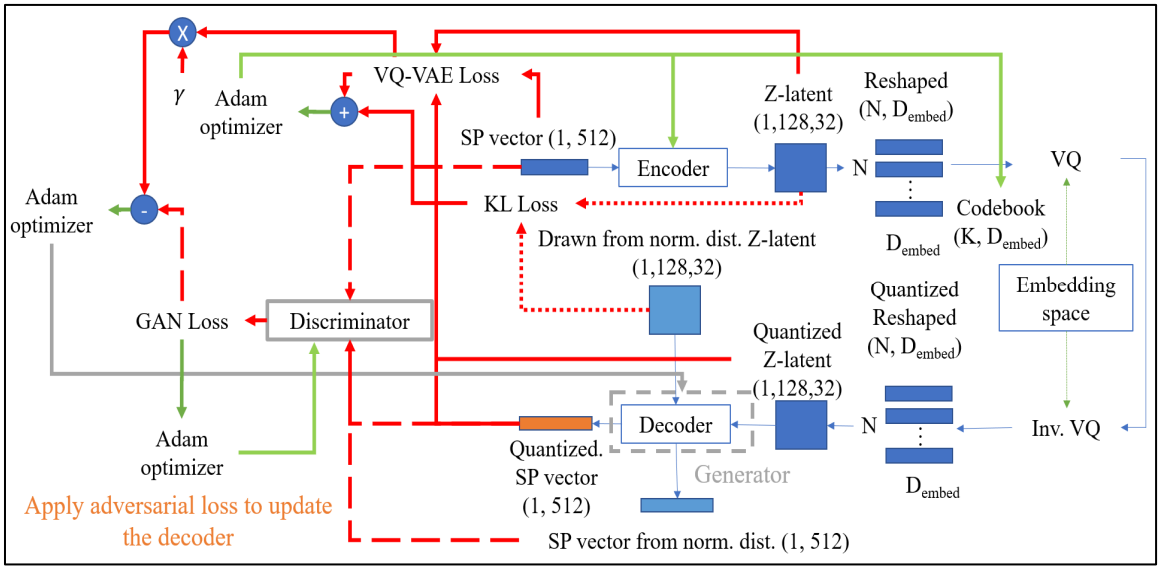

Figure 6.8: The training process of the VAEGAN implemented in VQ-VAE.

Figure 6.9 shows the training process of the VQ-VAE-EMGAN. The input SP vector is fed into the encoder network to produce the unquantized z-latent. It is reshaped and quantized with the embedding space. The index of the codebook of the embedding space is transmitted and reproduces the quantized z-latent by choosing the corresponding vector pattern in the embedding space as the input to the decoder network to reproduce the quantized SP vector. The first loss term (at line 11, Figure 6.2) in Equation 6.1 is applied to calculate the loss to update the encoder network parameters and decoder network parameters with the Adam optimizer in Equation 6.8. The discriminator network is organized to discriminate between the unquantized and the quantized z-latents. The generator loss is computed by Equation 6.28, and the discriminator loss is computed by Equation 6.29. The adversarial loss from the discriminator is used at the Adam optimizer in Equation 6.8 to update the embedding space parameters assumed as the generator network. The adversarial loss from the discriminator is utilized to update discriminator network parameters with the Adam optimizer in Equation 6.8.

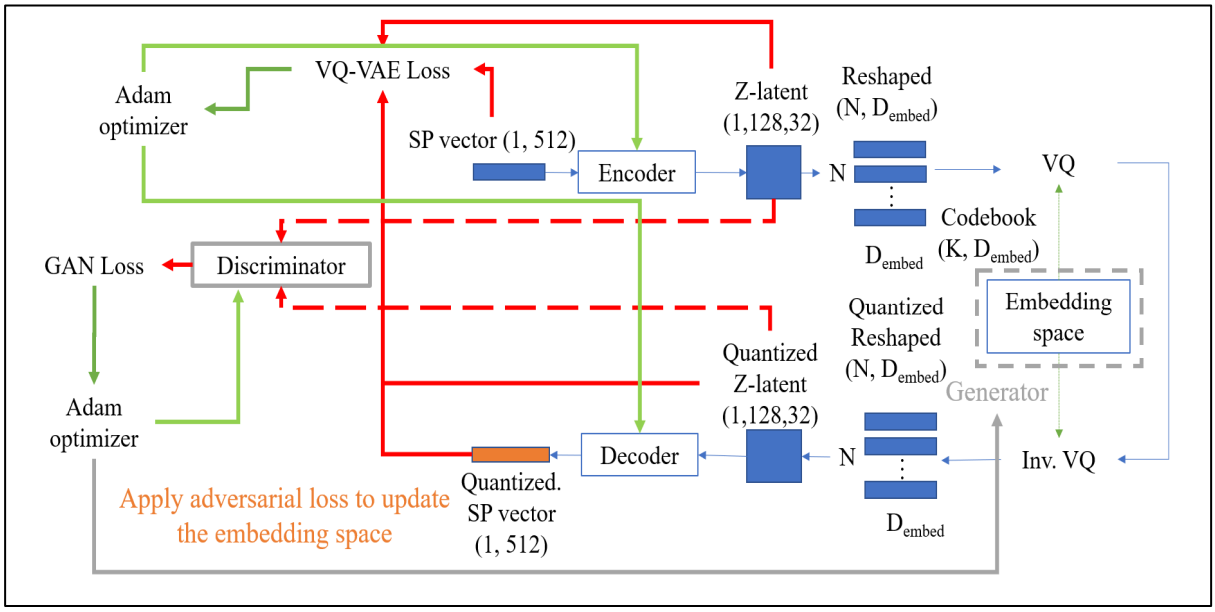

Figure 6.9: The VQ-VAE-EMGAN training process.

Figure 6.10 shows the training process of the VQ-VAE-EMDEC. The input SP vector is fed into the encoder network to produce the unquantized z-latent. It is reshaped and quantized with the embedding space. The index in the codebook of the embedding space is transmitted and reproduces the quantized z-latent by choosing the corresponding vector pattern in the embedding space as the input to the decoder network to reproduce the quantized SP vector. The VQ-VAE-EMDEC technique is the same as the VAEGAN implemented in the VQ-VAE, which draws the sample from a normal distribution with the same shape as the z-latent and then feeds into the decoder network to produce the SP vector from the normal distribution for calculating the KL divergence loss in Equation 6.30. The second loss term L (at line 14, Figure 6.3), defined by Equation 6.1, calculates the loss with the Adam optimizer in Equation 6.8. The discriminator network is organized to discriminate between the unquantized and the quantized SP vectors. Equation 6.31 can calculate the adversarial loss. The loss term updates the encoder network parameters with the Adam optimizer presented in Equation 6.32. The decoder network parameters and embedding space parameters updated by Adam optimizer from the loss term in Equation 6.33, γ is a weight set to 10. The discriminator network parameters are updated by Equation 6.34 with the Adam optimizer in Equation 6.8.

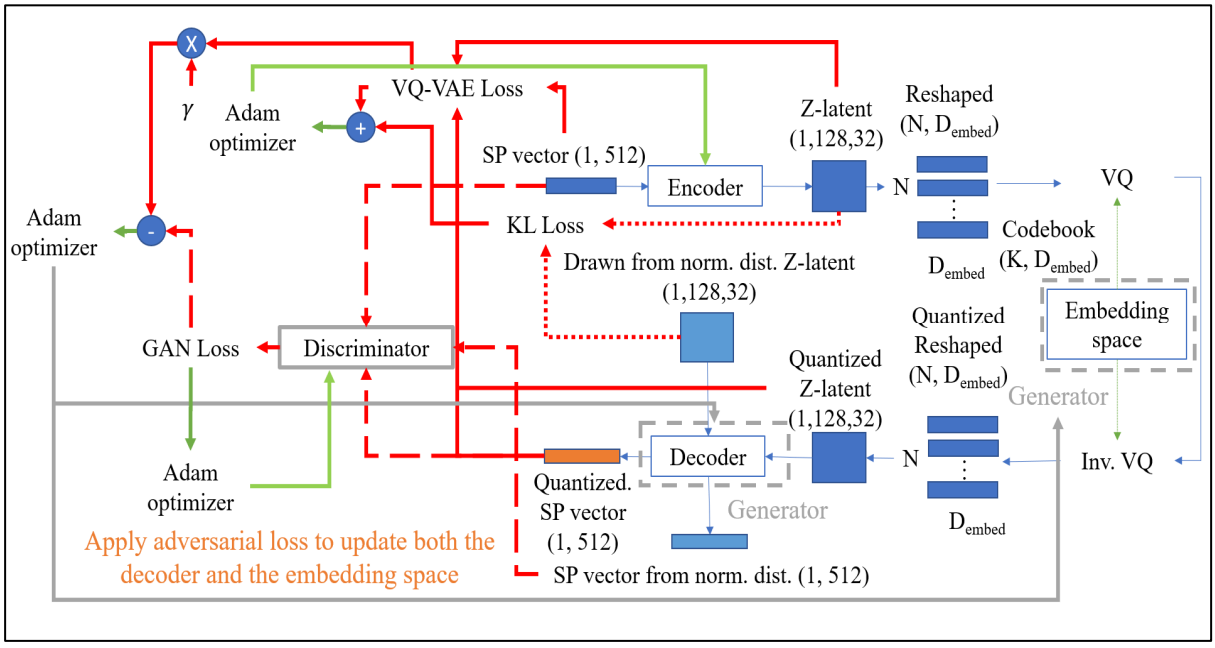

Figure 6.10: The VQ-VAE-EMDEC training process.

6.3.3 Implementation of the VQ-VAE, VAEGAN implemented in VQ- VAE, the VQ-VAE-EMGAN, and the VQ-VAE-EMDEC

For the based adversarial technique models, VAEGAN implemented in VQ-VAE, VQ-VAE-EMGAN, and the VQ-VAE-EMDEC, the discriminator network organizes another network architecture shown in Table 6.2. For the input layer of the discriminator network, the fully-connected input layer of the proposed VQ-VAE-EMGAN is different from VAEGAN implemented in VQ-VAE, and the VQ-VAE-EMDEC, the number of neurons which is determined by the received z-latent shape of input data of the models. The input layer transforms the input data into the fitted size of the following convolutional layers to calculate the sum into a single constant for adversarial loss calculation.

Table 6.2
The discriminator network architecture of the VAEGAN implemented in VQVAE, the VQ-VAE-EMGAN, and the VQ-VAE-EMDEC.

Discriminator network	
Layer	Architecture
Input_1	512 FC, LeakyReLU, Dropout (VAEGAN implemented in VQ-VAE) 128 FC, LeakyReLU, Dropout (VQ-VAE-EMGAN) 512 FC, LeakyReLU, Dropout (VQ-VAE-EMDEC)
Hidden_1	2×2 8 Conv, LeakyReLU, Dropout, BatchNorm
Hidden_2	2×2 16 Conv, LeakyReLU, Dropout, BatchNorm
Hidden_3	2×2 32 Conv, LeakyReLU, Dropout, BatchNorm
Output	2×2 64 Conv, Sigmoid

The switching of the discriminator adversarial loss is different among the three methods for optimizing specific network parameters. The proposed VAEGAN implemented in VQ-VAE utilizes the adversarial loss to update the decoder network parameters. The proposed VQ-VAE-EMGAN focuses to update the embedding space network parameters by the adversarial loss. The proposed VQ-VAE-EMDEC is different from the two proposed methods. The adversarial loss is utilized to update both decoder network parameters and the embedding space parameters.

In the experiments, the four bitrate models of each the VQ-VAE, the VAEGAN implemented in VQ-VAE, the VQ-VAE-EMGAN, and the VQ-VAE-EMDEC were constructed with the same four bitrates for the performance comparison, as presented in Table 6.3. The operational bitrates were 128, 256, 512, and 1024 bit/SP vector for the first, second, third, and fourth models of the VQ-VAE, the VAEGAN implemented in VQ-VAE, the VQ-VAE-EMGAN, and the VQ-VAE-EMDEC. All sixteen models utilized 0.0001 as the learning rate in the training process. The mini-batch size was 32 spectral envelope frames. The number of epochs was two of raw speech waveform in the training dataset. The Adam optimization was applied to update all the network parameters.

Table 6.3
The four implementation models for the VQ-VAE, the VAEGAN implemented in VQ-VAE, the VQ-VAE-EMGAN, and the VQ-VAE-EMDEC.

Model	z-latent (H, W, D)	Reshaped z-latent (N, D_{embed})	Embedding Space (K, D_{embed})	$\frac{\text{Bits}}{\text{SP vector}}$
1	(1, 128, 32)	(128, 32)	(256, 32)	1024
2		(64, 64)	(256, 64)	512
3		(32, 128)	(256, 128)	256
4		(16, 256)	(256, 256)	128

6.3.4 The results of the performance comparison

Experimental results utilized the Log Spectral Distortion (LSD) [85, 86] defined in Equation 6.24 as the spectral envelope distortion indicator. The L2 error was applied to measure the quality of the z-latent error between z-latent and quantized z-latent, as defined in Equation 6.25. The PESQ [87] was utilized to evaluate the quality between the original raw speech waveform and the reconstructed speech waveform.

Figures 6.11, 6.12, 6.13, and 6.14 shows the calculation process of the average LSD, the z-latent L2 error, and the PESQ score for the VQ-VAE, the proposed VAEGAN implemented in VQ-VAE, the VQ-VAE-EMGAN, and the VQ-VAE-EMDEC, respectively. The testing dataset consisted of 100 raw speech waveforms from the LibriSpeech corpus that were not included in the training process of the proposed VAEGAN implemented in VQ-VAE, VQ-VAE-EMGAN, and the VQ-VAE-EMDEC. The WORLD vocoder extracted the speech parameters from each raw speech waveform, only the spectral envelope parameter (SP) was quantized, and the synthesis process reconstructed the raw speech waveform again based on the quantized SP and other parameters. The LSD was applied to measure the distortion between SP and quantized SP. The L2 error was applied to measure the quality of the z-latent error between z-latent and quantized z-latent. Finally, the 100 LSD values, z-latent L2 errors, and PESQ scores were calculated for each waveform and the average of LSD, z-latent L2, and PESQ scores were obtained.

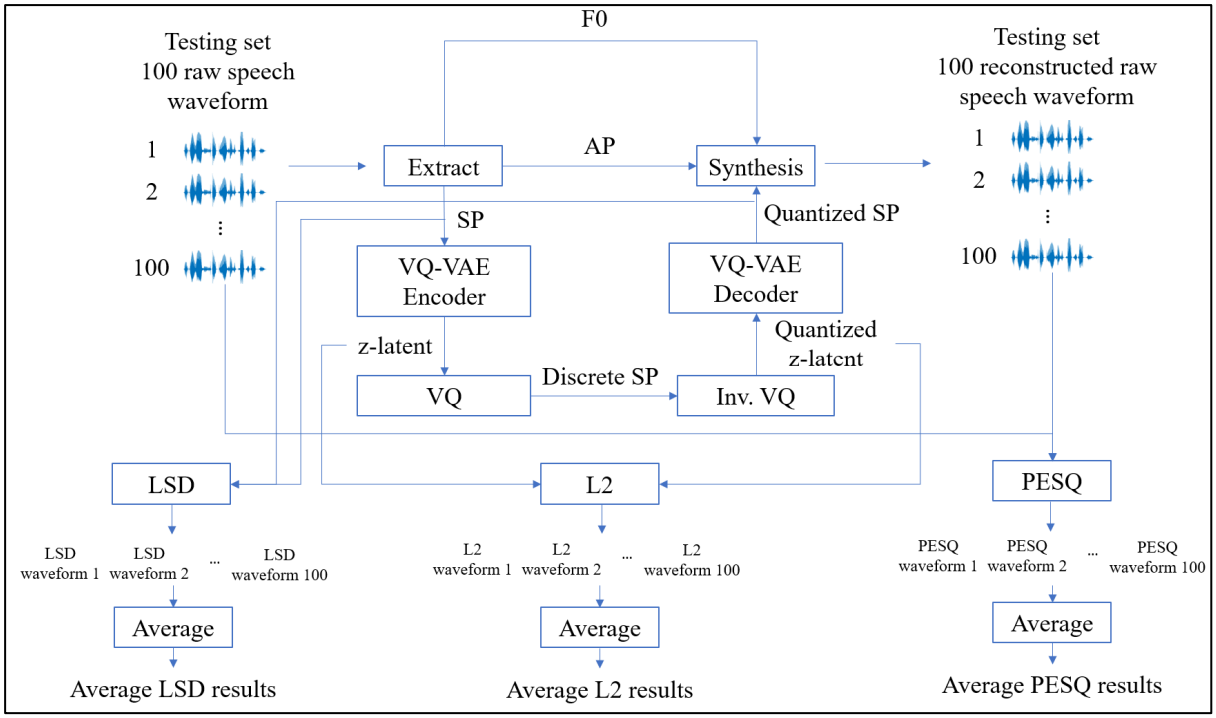

Figure 6.11: The VQ-VAE average of LSD, z-latent L2 error, and the PESQ score evaluation.

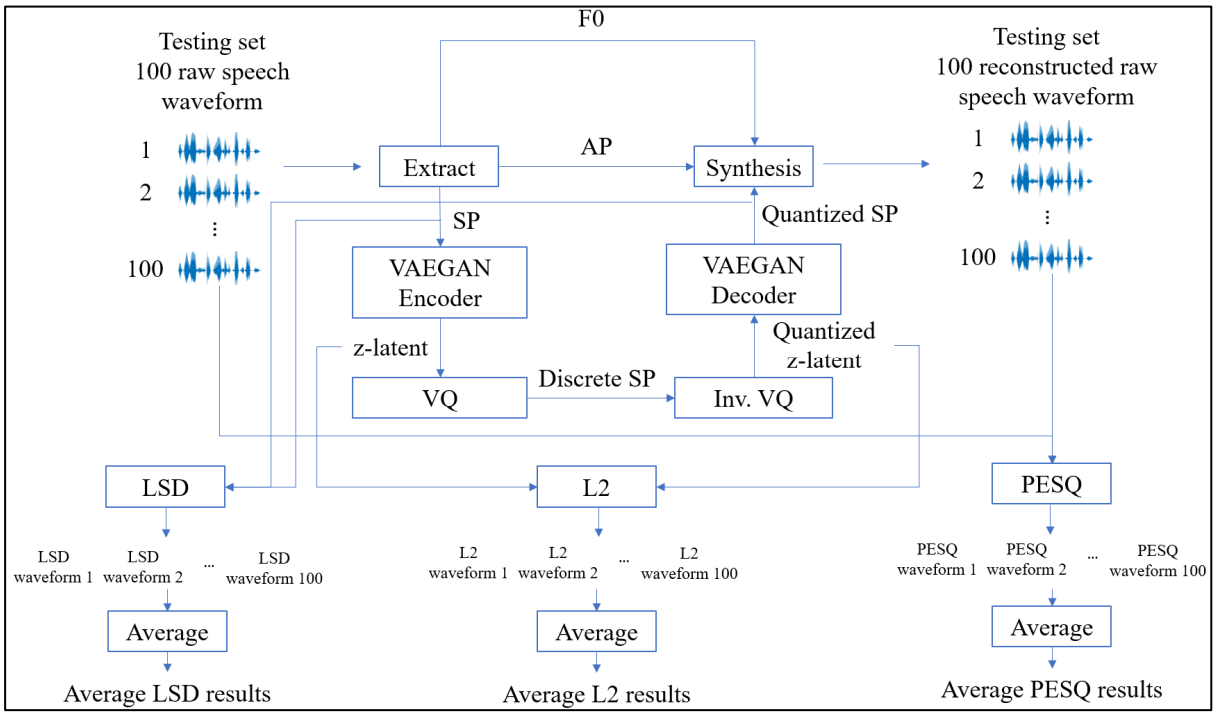

Figure 6.12: The proposed VAEGAN implemented in VQ-VAE average of LSD, z-latent L2 error, and the PESQ score evaluation.

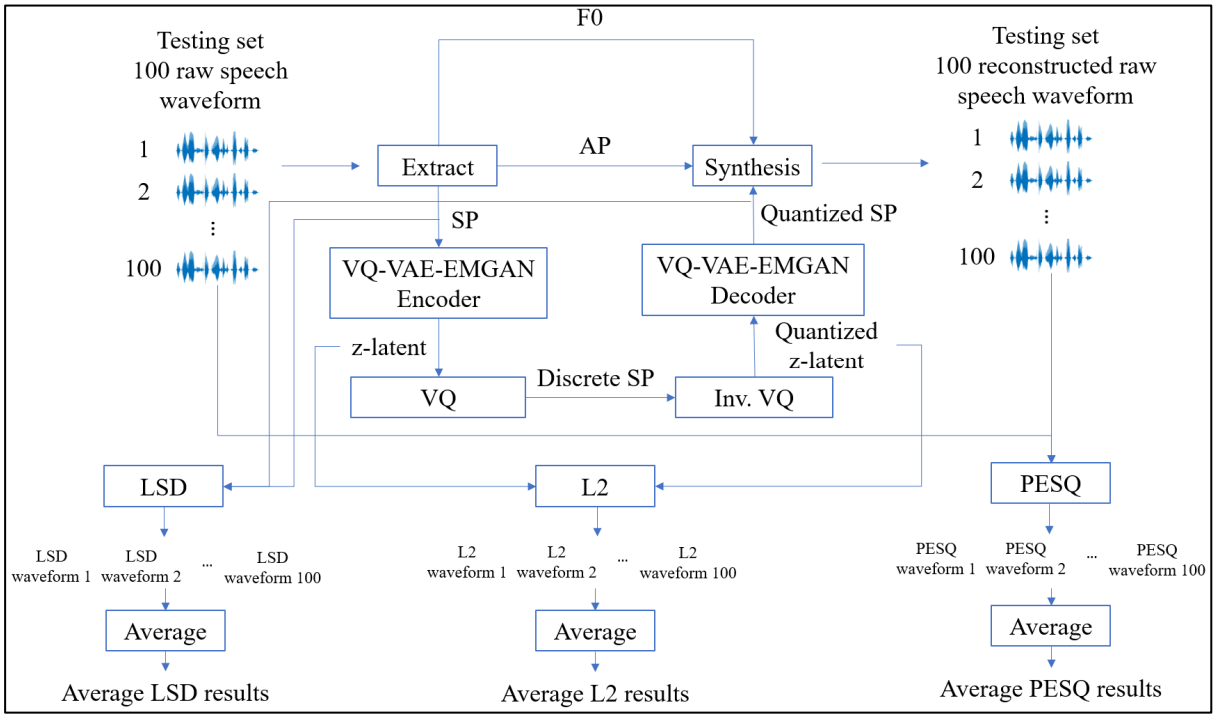

Figure 6.13: The proposed VQ-VAE-EMGAN average of LSD, z-latent L2 error, and the PESQ score evaluation.

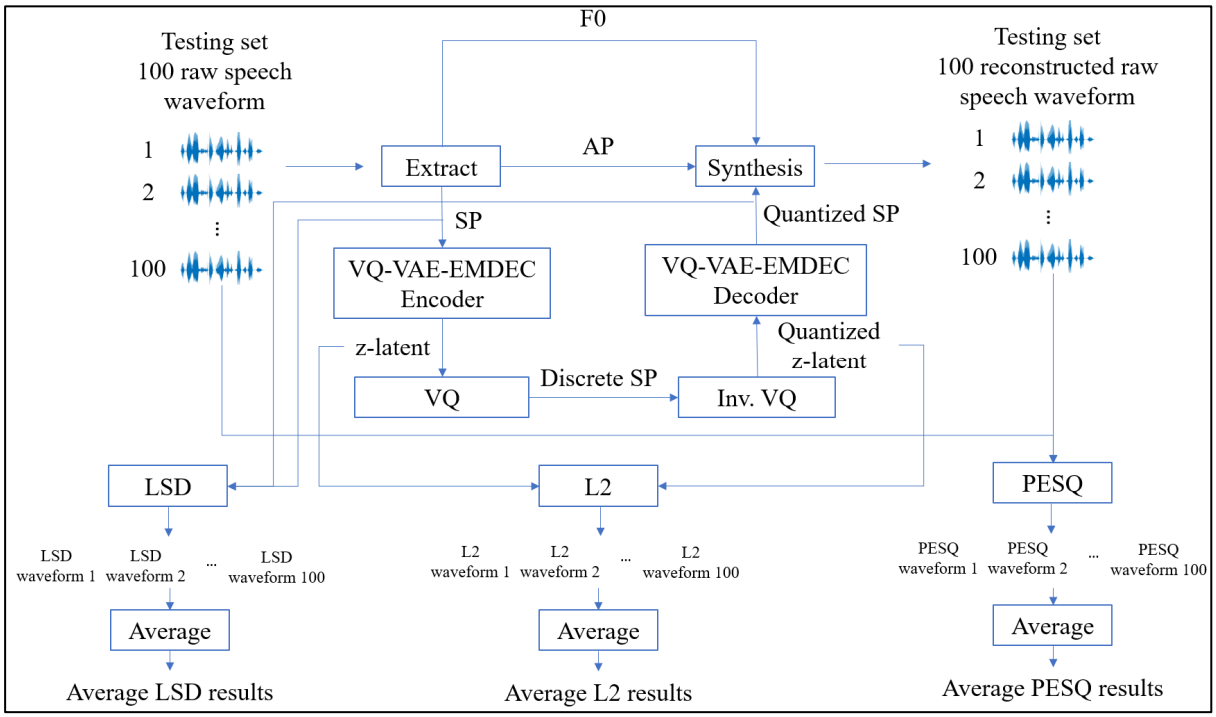

Figure 6.14: The proposed VQ-VAE-EMDEC average of LSD, z-latent L2 error, and the PESQ score evaluation.

All sixteen models were trained with the training set raw speech waveforms by extracted spectral envelopes from the WORLD vocoder and trained in the fixed environment. The spectral envelope quantization performance results were evaluated by the Log Spectral Distortion (LSD) in Equation 6.38 to measure the differences between the unquantized spectral envelope frame (SP vector) and quantized spectral envelope frame (SP_q vector) in Figure 6.4, and the L2 loss in Equation 6.39 for the four models of VQ-VAE, VAEGAN implemented in VQ-VAE, VQ-VAE-EMGAN, and VQ-VAE-EMDEC. The LSD in dB and the L2 are defined as followings:

$$LSD_{db} = 10 \times \frac{2}{M} \sqrt{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (X_{ij} - Y_{ij})^2}, \tag{6.38}$$

where M is the number of log-spectral coefficients frames, N is the length of log-spectral coefficients, X_{ij} is the logarithm base ten spectral coefficients of unquantized spectral envelope, and Y_{ij} is the logarithm base ten log-spectral coefficients of the quantized spectral envelope.

$$L2 = \sum_{i=1}^{n} \sum_{j=1}^{m} (z_e(x)_{ij} - z_q(x)_{ij})^2, \tag{6.39}$$

where m is the number of frames, N is the length of a frame vector, $z_e(x)$ is the unquantized z-latent, and $z_q(x)$ is the quantized z-latent.

The Perceptual Evaluation of Speech Quality (PESQ) standard as ITU-T recommendation [87] was applied to measure the quality of the reconstructed 16 kHz waveform of WORLD vocoder for the VQ-VAE and proposed three methods for spectral envelope quantization with the regular WORLD vocoder without any spectral envelope quantization.

Table 6.4 and Figures 6.15, 6.16, 6.17, and 6.18 show the model comparison results for the methods in terms of the average LSD, average L2, and average PESQ results. In Table 6.4, the z-latent is represented with the shape of height, width, and depth as (H, W, D), and the reshaped z-latent shape as the number of vectors to be quantized with the length of the vector (N, D_{embed}). The embedding space size is the number of vectors with the length of the vector (K, D_{embed}). The bitrates of each input SP vector are represented as Bits/SP vector. In the figures, the models with adversarial training on the embedding space, VQ-VAE-EMGAN and VQ-VAE-EMDEC decreased the LSD and L2 compared with the reference VQ-VAE method. They also increased the PESQ scores compared with the VQ-VAE.

Table 6.4
The comparison results.

VQ-VAE							
Model	z-latent (H, W, D)	Reshaped z-latent (N, D_{embed})	Embedding Space (K, D_{embed})	z-latent error (L2)	$\frac{Bits}{SP}$	LSD	PESQ
1	(1, 128, 32)	(128, 32)	(256, 32)	0.21	1024	3.28	2.53
2		(64, 64)	(256, 64)	0.30	512	3.89	2.35
3		(32, 128)	(256, 128)	0.39	256	4.92	2.11
4		(16, 256)	(256, 256)	0.67	128	7.92	1.55
VAEGAN implemented in VQ-VAE							
Model	z-latent (H, W, D)	Reshaped z-latent (N, D_{embed})	Embedding Space (K, D_{embed})	z-latent error (L2)	$\frac{Bits}{SP}$	LSD	PESQ
1	(1, 128, 32)	(128, 32)	(256, 32)	0.17	1024	2.39	3.28
2		(64, 64)	(256, 64)	0.33	512	4.77	2.22
3		(32, 128)	(256, 128)	0.44	256	6.09	1.61
4		(16, 256)	(256, 256)	0.71	128	9.28	1.15
VQ-VAE-EMGAN							
Model	z-latent (H, W, D)	Reshaped z-latent (N, D_{embed})	Embedding Space (K, D_{embed})	z-latent error (L2)	$\frac{Bits}{SP}$	LSD	PESQ
1	(1, 128, 32)	(128, 32)	(256, 32)	0.11	1024	2.85	2.78
2		(64, 64)	(256, 64)	0.28	512	3.67	2.46
3		(32, 128)	(256, 128)	0.37	256	5.04	2.29
4		(16, 256)	(256, 256)	0.56	128	6.57	1.70
VQ-VAE-EMDEC							
Model	z-latent (H, W, D)	Reshaped z-latent (N, D_{embed})	Embedding Space (K, D_{embed})	z-latent error (L2)	$\frac{Bits}{SP}$	LSD	PESQ
1	(1, 128, 32)	(128, 32)	(256, 32)	0.13	1024	2.21	3.19
2		(64, 64)	(256, 64)	0.16	512	2.60	3.01
3		(32, 128)	(256, 128)	0.34	256	4.68	2.12
4		(16, 256)	(256, 256)	0.51	128	6.60	1.51

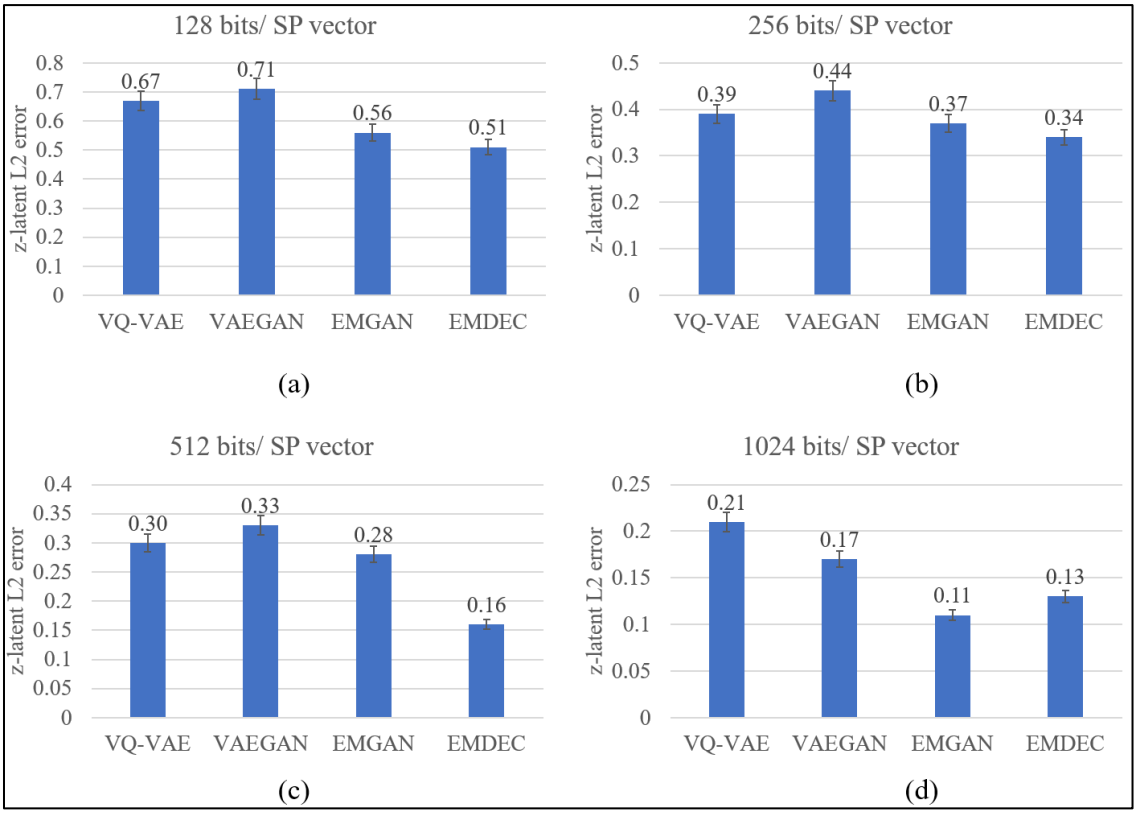

Figure 6.15: The varied bit/SP vector comparison results of z-latent L2 error.

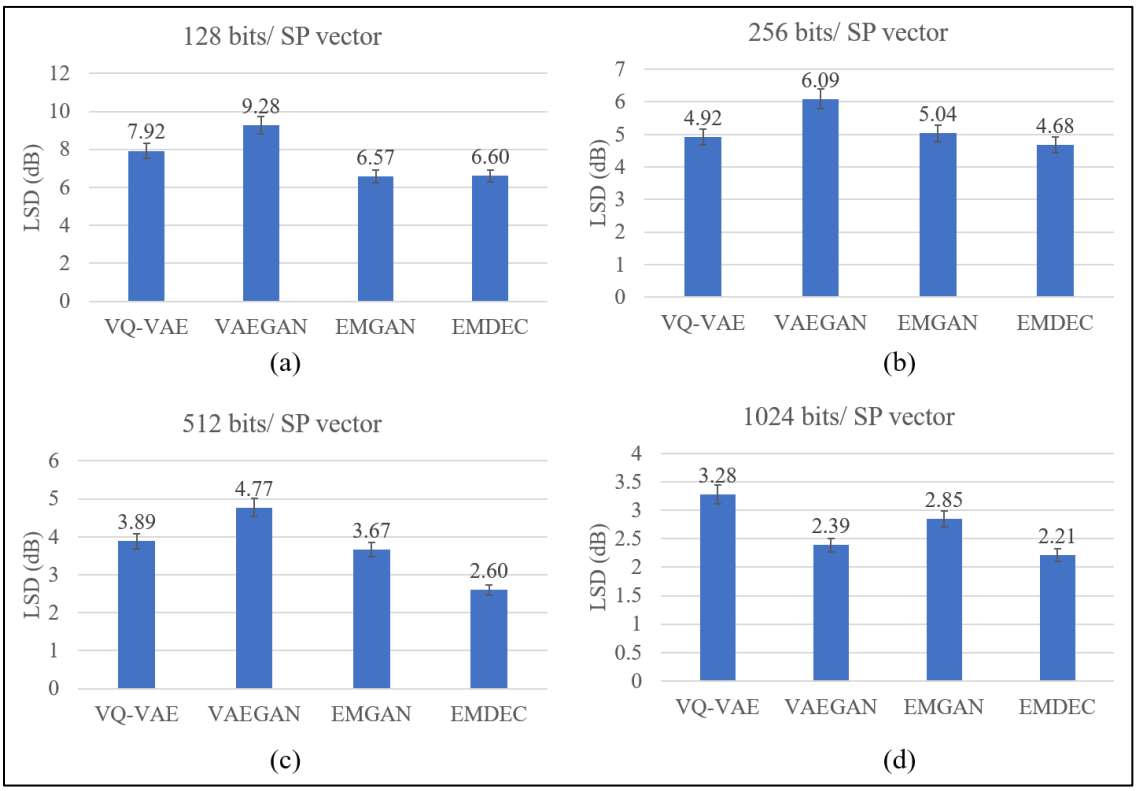

Figure 6.16: The varied bit/SP vector comparison results of LSD.

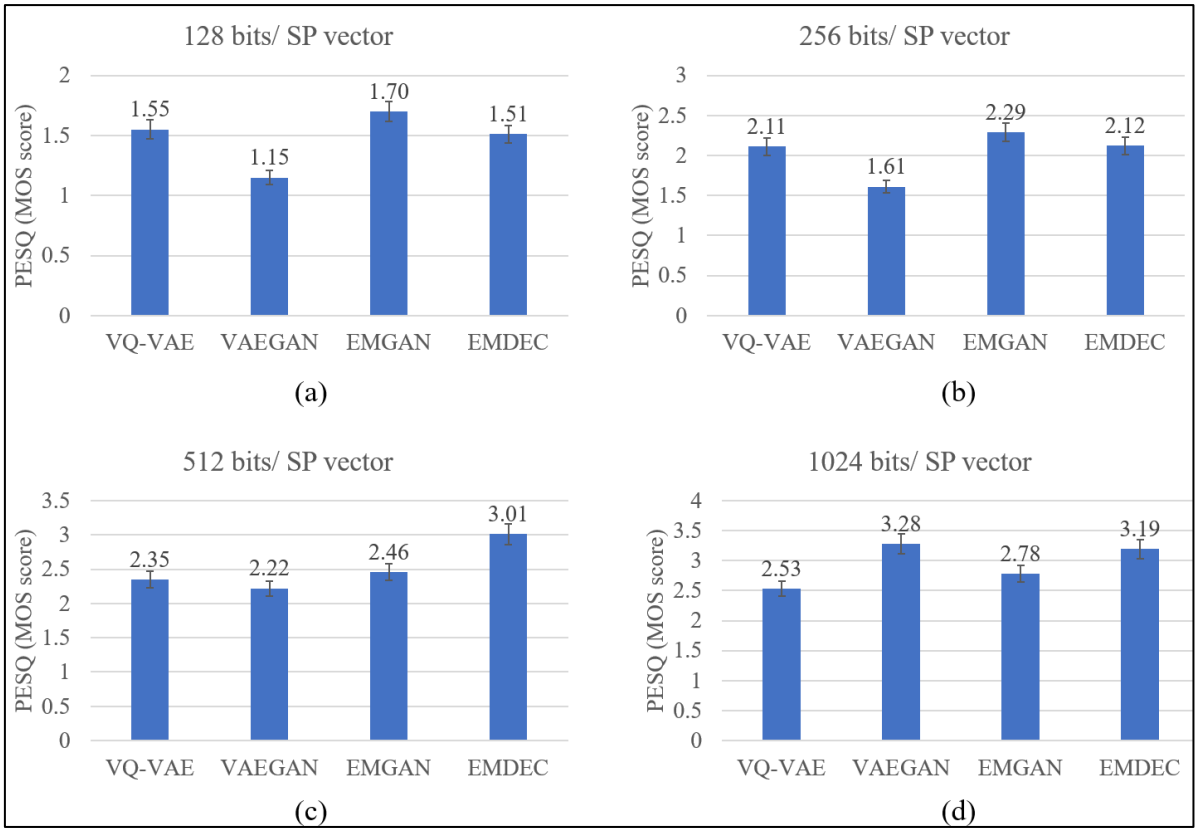

Figure 6.17: The varied bit/SP vector comparison results of PESQ.

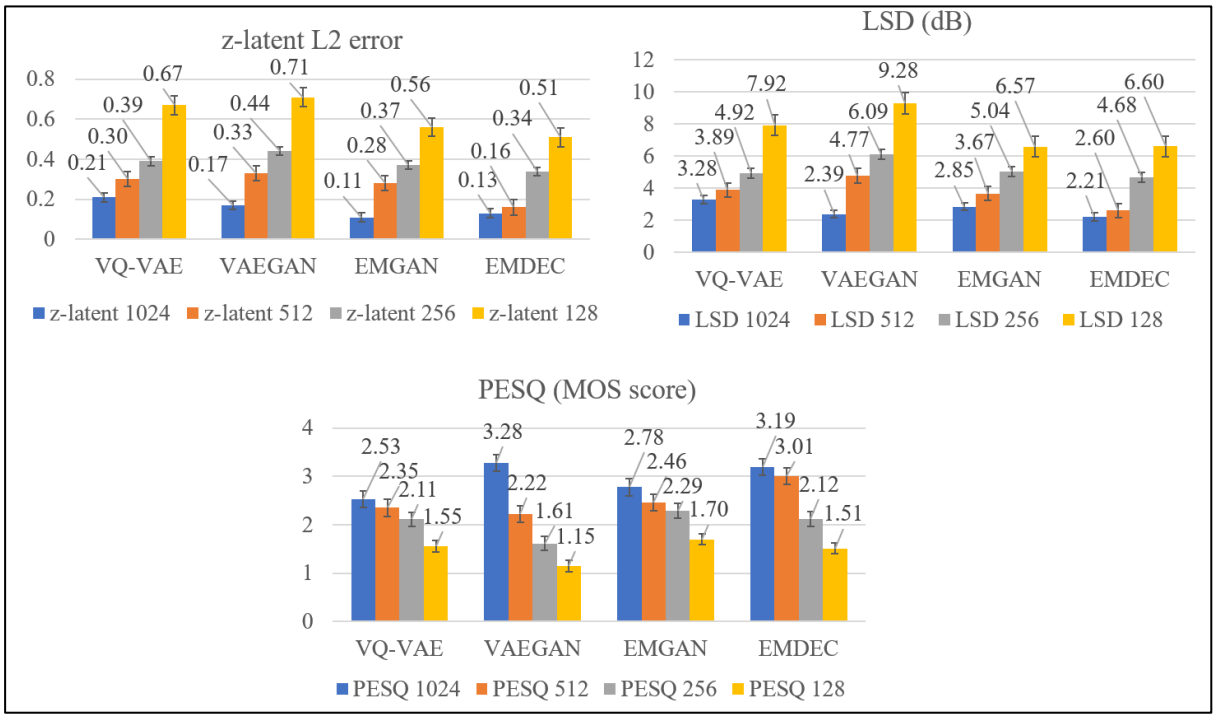

Figure 6.18: The loss term comparison results.

Figure 6.19 compares the examples of z-latents between unquantized and quantized methods at the operation bitrate of 1024 Bits/SP. The unquantized z-latents are different for each method due to the different encoder and decoder training. From the figure, the proposed techniques, the VQ-VAE-EMGAN and VQ-VAE-EMDEC give a clearer quantized z-latent compared to the VQ-VAE and the VAEGAN implemented in VQ-VAE. However, the results showed that the model such as the VAEGAN implemented in VQ-VAE using the adversarial loss updating only the decoder network could not operate well at the lower bitrate operation.

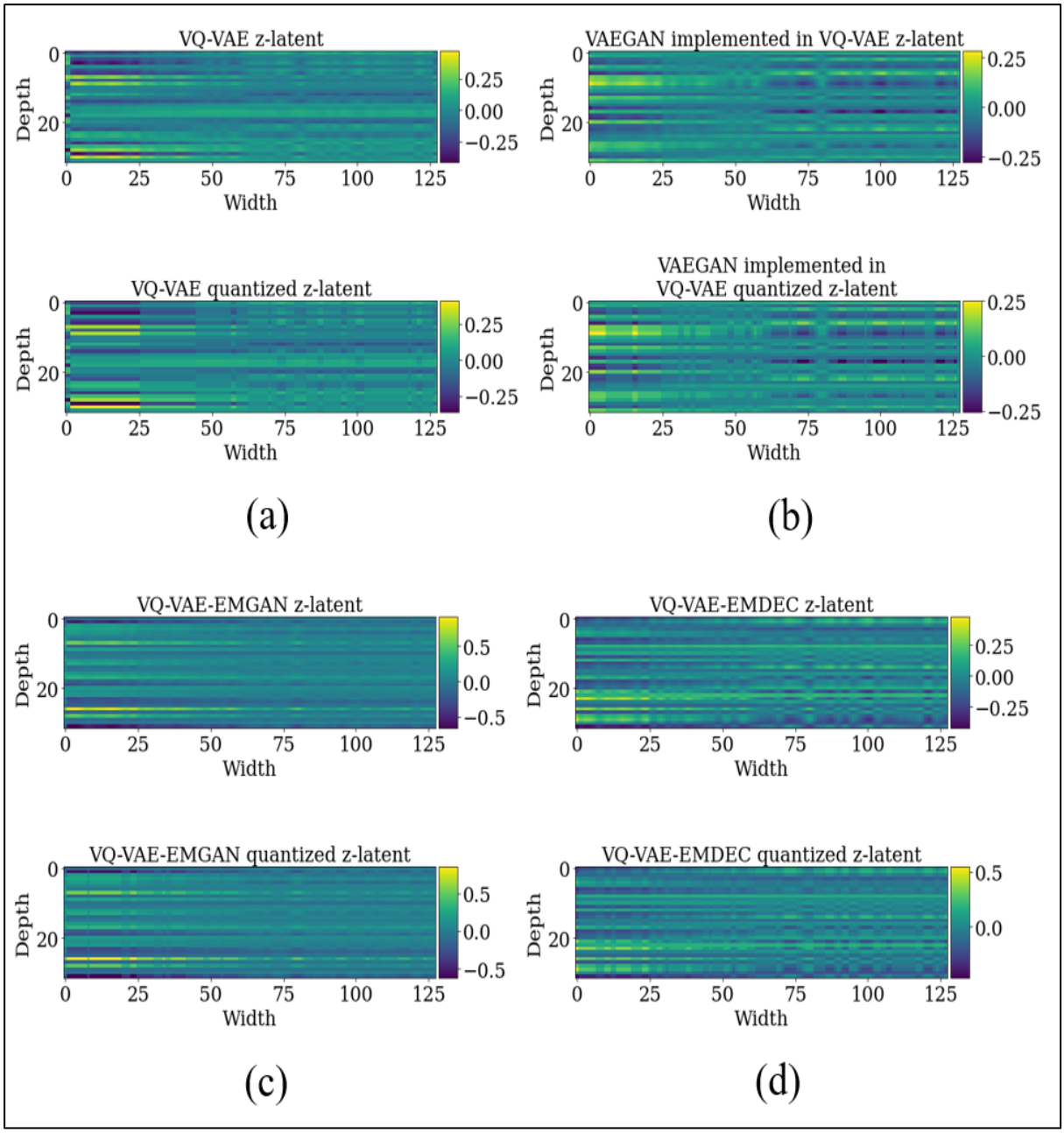

Figure 6.19: The sampled z-latent comparison between unquantized and quantized methods at the operation bitrate of 1024 Bits/SP.

Figures 6.20, 6.21, 6.22, and 6.23 compare the sampled WORLD vocoder phoneme spectral envelope frames for four bits operations.

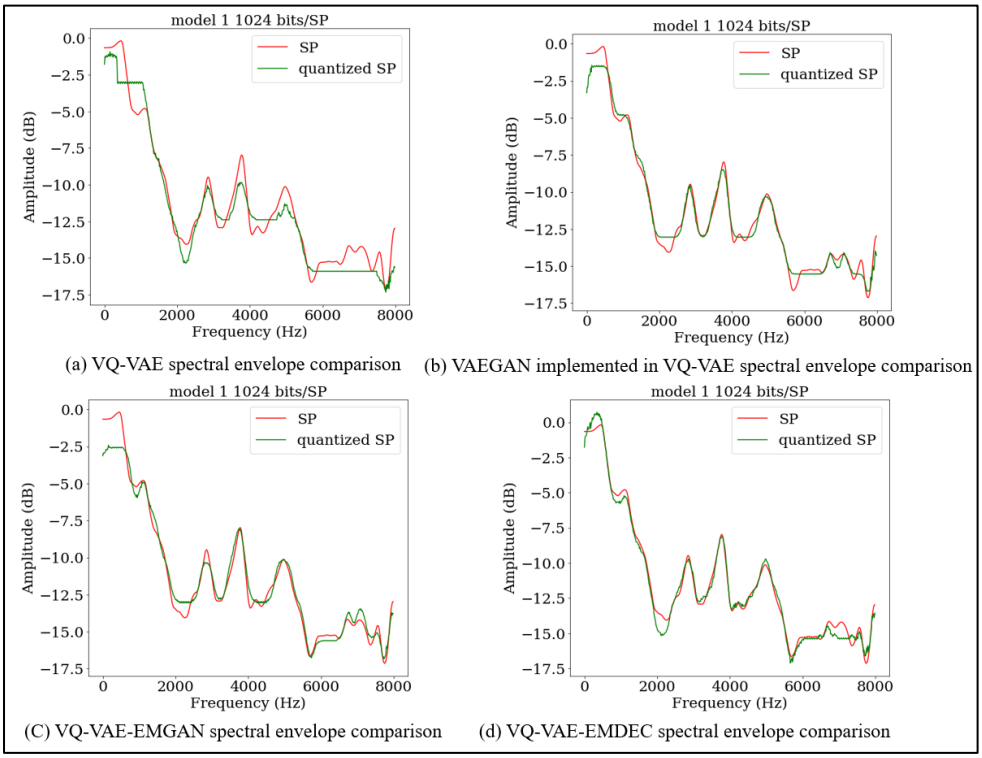

Figure 6.20: The sampled WORLD vocoder phoneme spectral envelope frames comparison 1024 bits/SP.

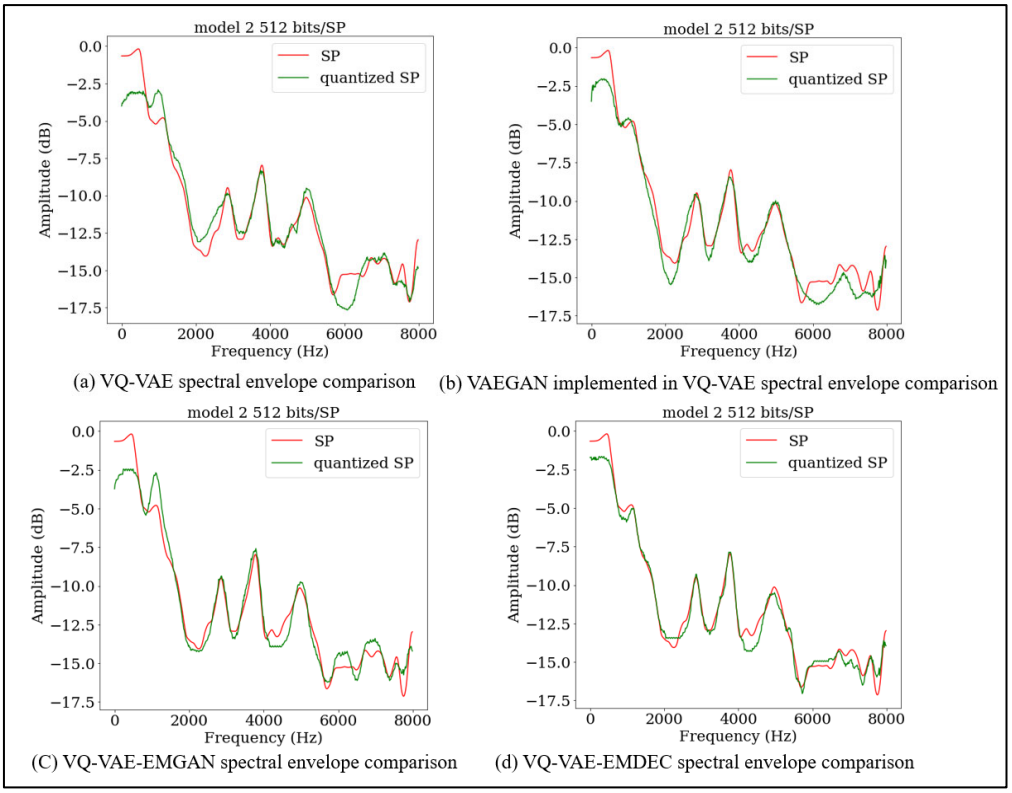

Figure 6.21: The sampled WORLD vocoder phoneme spectral envelope frames comparison 512 bits/SP.

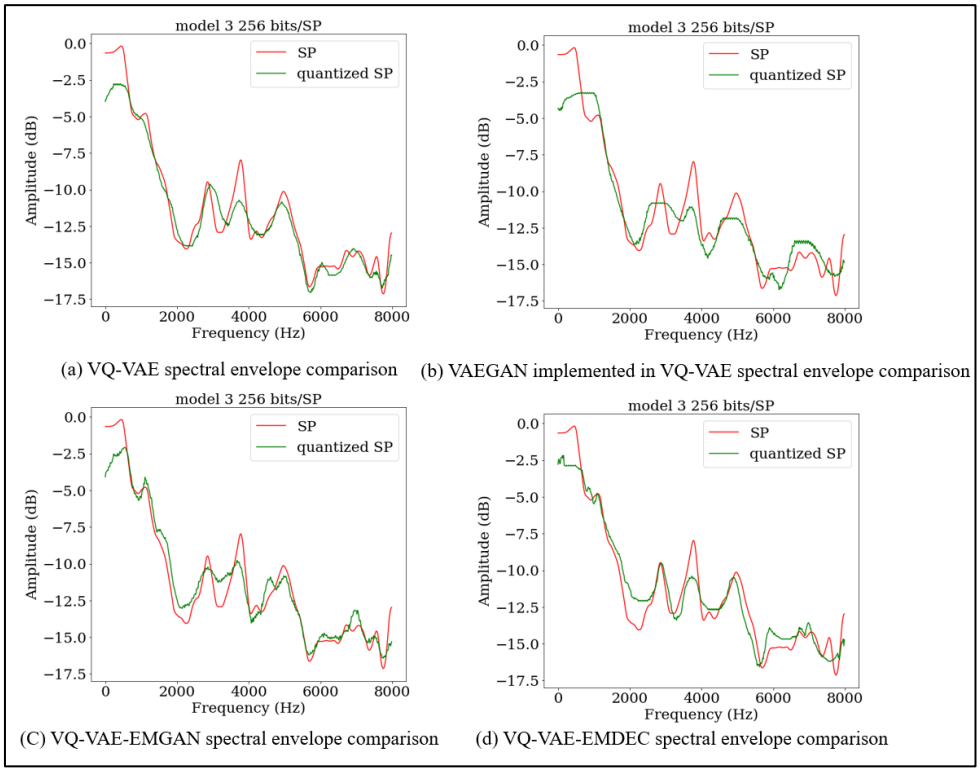

Figure 6.22: The sampled WORLD vocoder phoneme spectral envelope frames comparison 256 bits/SP.

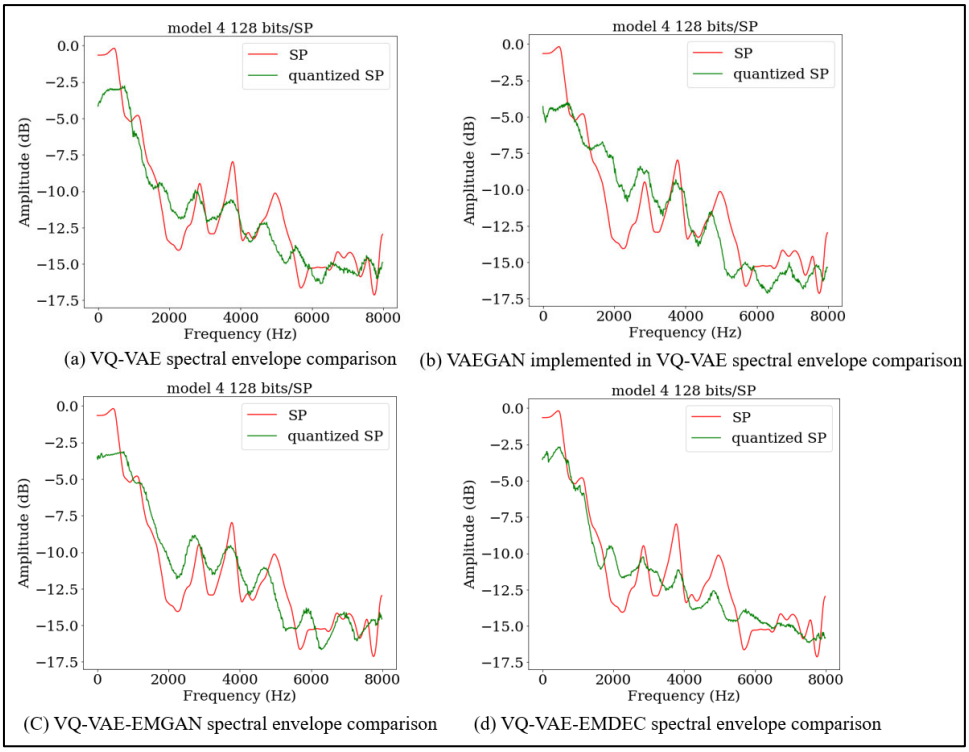

Figure 6.23: The sampled WORLD vocoder phoneme spectral envelope frames comparison 128 bits/SP.

Figures 6.24, 6.25, 6.26, and 6.27 show the sampled WORLD vocoder spectral envelope (spectrogram) with four bitrate operations.

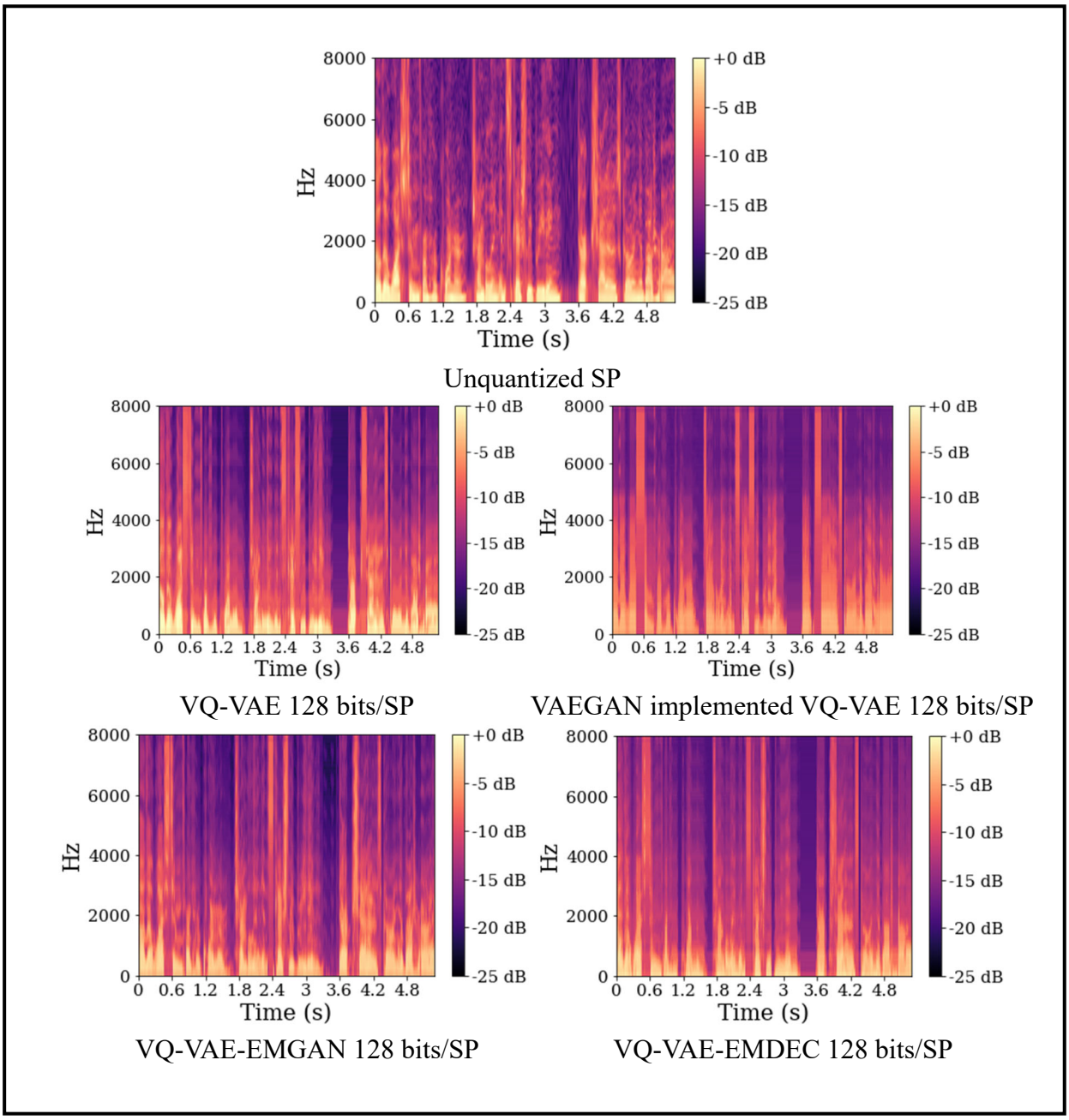

Figure 6.24: The sampled WORLD vocoder spectral envelope (spectrogram) at 128 bits/SP.

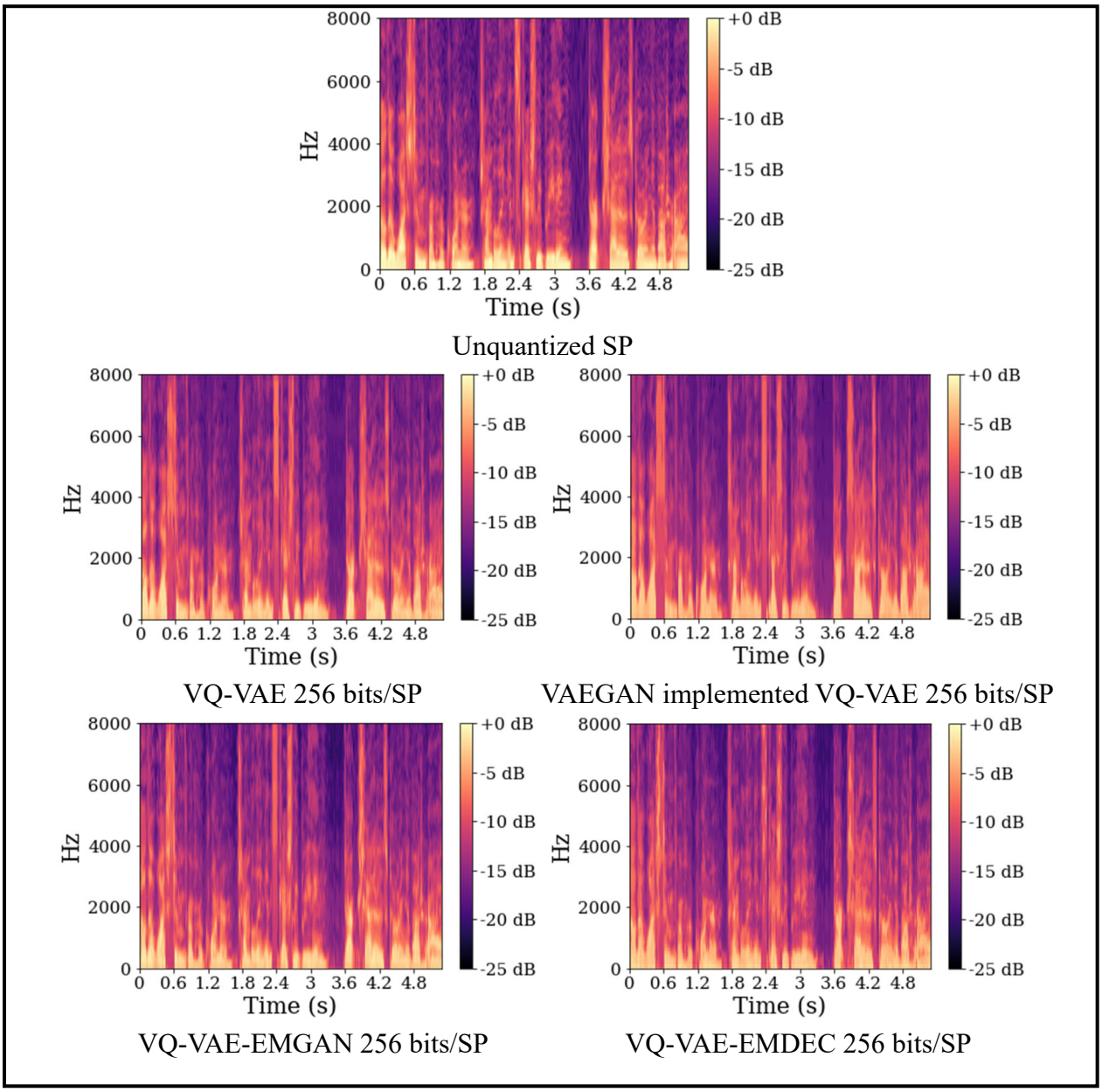

Figure 6.25: The sampled WORLD vocoder spectral envelope (spectrogram) at 256 bits/SP.

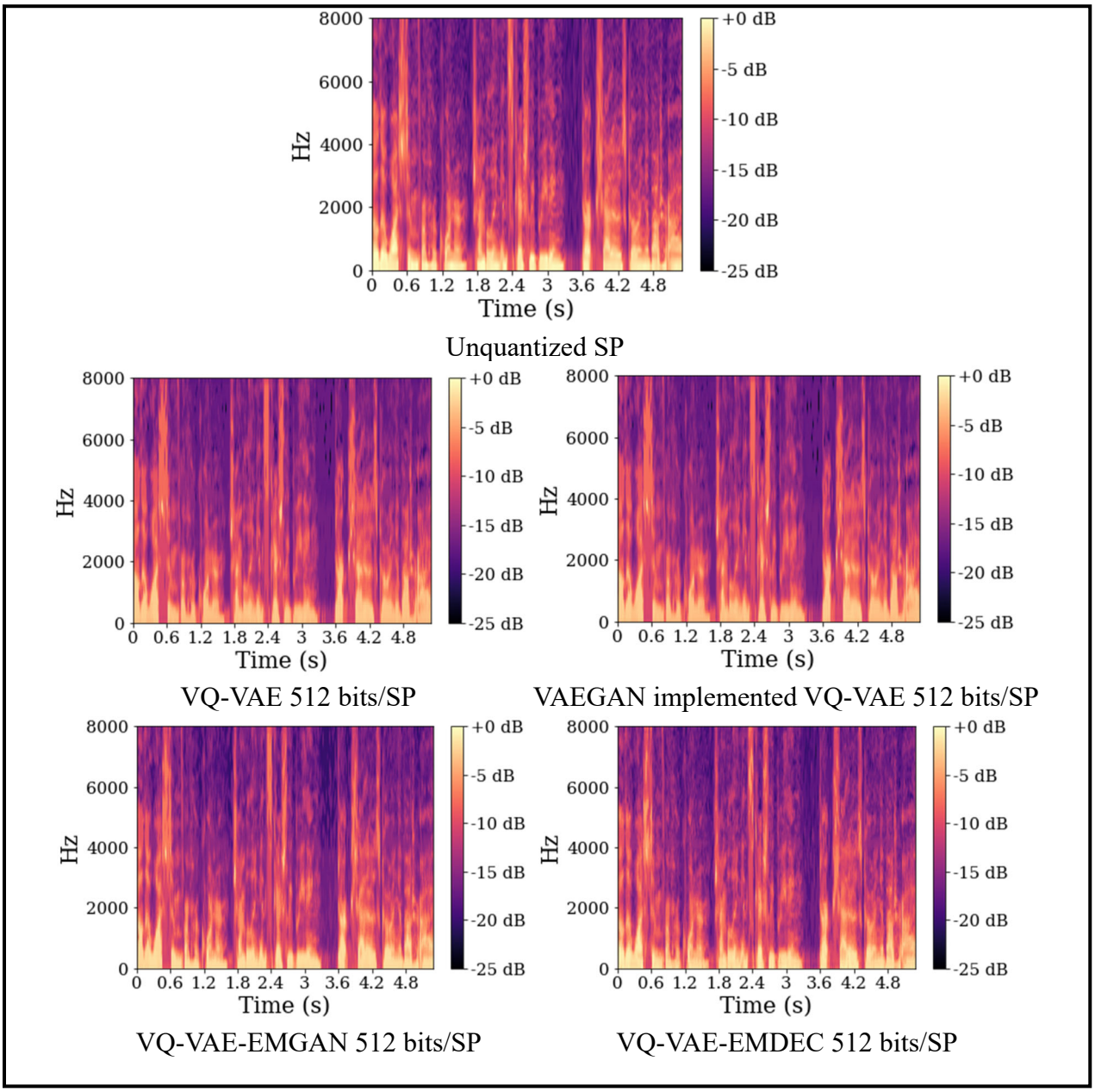

Figure 6.26: The sampled WORLD vocoder spectral envelope (spectrogram) at 512 bits/SP.

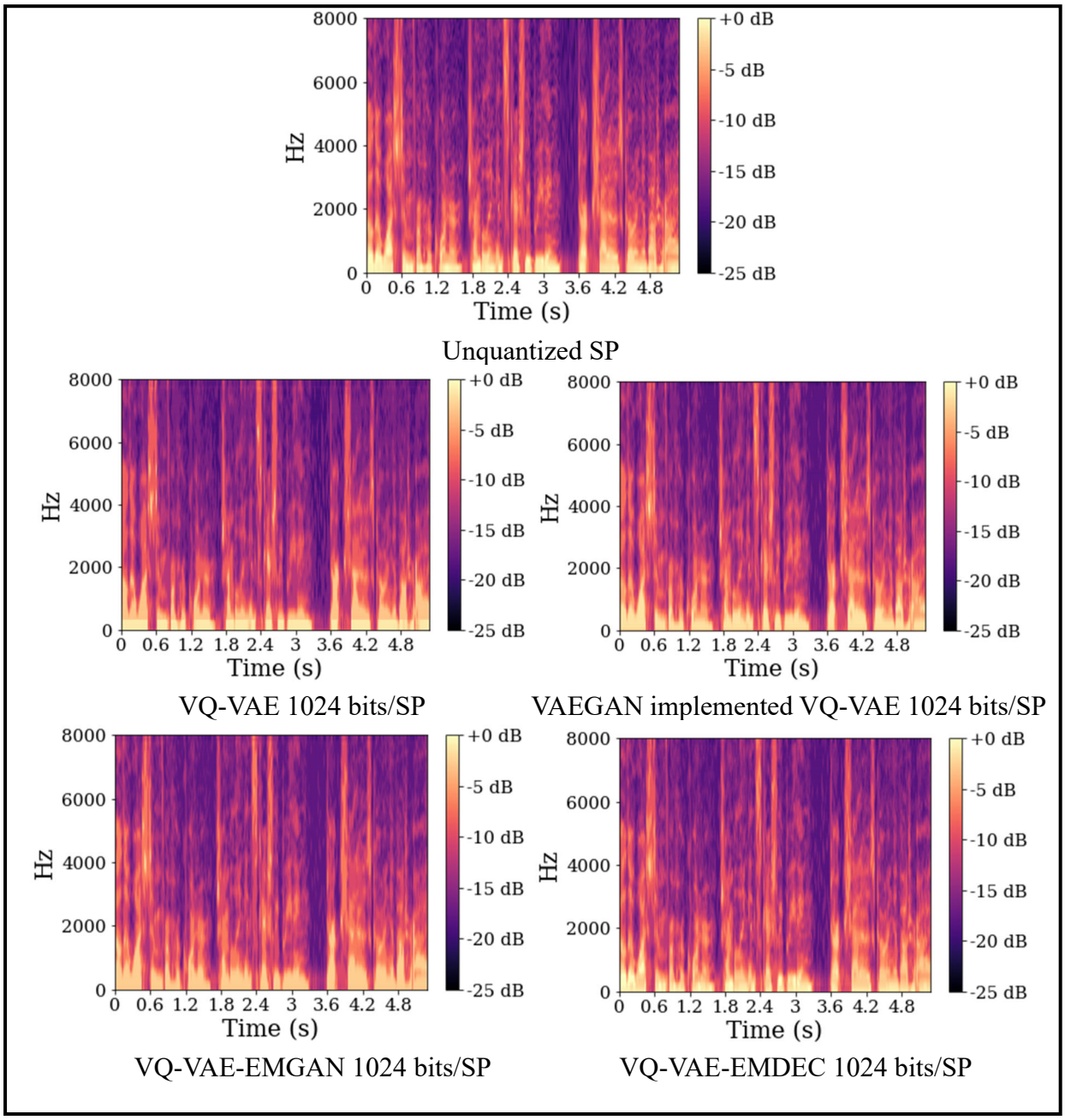

Figure 6.27: The sampled WORLD vocoder spectral envelope (spectrogram) at 1024 bits/SP.

The proposed VAEGAN implemented in VQ-VAE results obtained the best average PESQ score in the highest bits operation. The model operating at the lower bits/SP vector (512, 256, 128 bits/SP vector) did not perform well because this model discarded the adversarial loss to update the embedding. The average z-latent L2 loss was very high compared to the VQ-VAE and other proposed methods.

The proposed VQ-VAE-EMGAN was designed to optimize the embedding space to get the lowest z-latent distortion (z-latent L2 error) by adversarial loss. However, the decoder network parameters were discarded to apply the adversarial loss to update, and the average LSD results and the average PESQ were not the best in the experiment.

The proposed VQ-VAE-EMDEC was the combined idea of mixing the first proposed method and the second proposed method. Instead, to update a single network parameter, the VQ-VAE-EMDEC utilized the adversarial loss to update both embedding space parameters and the decoder network parameters. The results from the experiment indicated that the average z-latent L2 error, the average LSD, and the average PESQ score were improved from the second proposed method (VQ-VAE-EMGAN) at the first two bitrates (1024 and 512 bits/SP vector), but in the lower two bitrates (256, 128 bits/SP vector) results were worse than the proposed VQ-VAE-EMGAN.

6.3.5 The effects of the model parameter initialization

The model parameter initialization behaviors were also investigated in this study. The VQ-VAE-EMGAN model 1 was chosen to evaluate the effect of model parameter initializations. The first model parameter initialization behaviors experiment was the model initialization method. Four initialization methods were applied as the initialization model parameters: a normal distribution, Xavier (uniform distribution), and Xavier (normal distribution). The drawn values from distribution are presented in Equations 6.40, 6.41, 6.42, and 6.43, respectively.

$$w_{norm} = N(\mu, \sigma^2)\,; \mu = 0.0\ and\ \sigma = 1.0, \tag{6.40}$$

$$w_{uni} = U(a, b)\,; a = 0.0\ and\ b = 1.0, \tag{6.41}$$

$$w_{xavier\_uni} = U(-a, a); a = gain \times \sqrt{\frac{6}{fan_{in}+fan_{out}}}\ and\ , \tag{6.42}$$

$$w_{xavier\_norm} = N(\mu = 0, \sigma^2);\ \sigma = \ gain \times \sqrt{\frac{2}{fan_{in}+fan_{out}}}, \tag{6.43}$$

where $N(\mu, \sigma^2)$ is the normal distribution. μ is the mean, σ is the standard deviation. $U(a, b)$ is the uniform distribution. a is the lower bound and b is the upper bound of the distribution. $gain$ is the weight control parameter. The fan_{in} is the number of nodes in the previous layer. The fan_{out} is the number of nodes in the next layer.

Table 6.5 shows the effect of first model parameter initialization behaviors. The model parameters drawn from the normal distribution obtained the best results of LSD, z-latent error L2, and PESQ.

The second experiment of the model parameter initialization was the VQ-VAE-EMGAN model 1 model parameters initialized with normal distribution. In this experiment, we repeated the initialization process ten times to investigate the effect of initial values. The results are shown in Table 6.6 that each time of parameter initialization also effect to the model performance, but the results of each attempt were not significantly different after finishing the model training.

Table 6.5
The VQ-VAE-EMGAN model 1 initialization method comparison results

Initialization method	$\frac{Bits}{SP}$	z-latent error (L2)	LSD	PESQ
Norm.	1024	0.13	2.21	3.19
Uniform		1.72×10^{30}	69.96	1.11
Xavier Norm.		0.15	2.88	2.80
Xavier Uniform		0.49	6.42	2.27

Table 6.6
The VQ-VAE-EMGAN model 1 repeat model parameter initialization comparison results.

Attempt	$\frac{Bits}{SP}$	z-latent error (L2)	LSD	PESQ
1	1024	0.09	1.82	3.75
2		0.12	2.13	3.42
3		0.11	1.99	3.18
4		0.15	2.55	3.38
5		0.14	2.35	3.42
6		0.15	2.56	3.15
7		0.22	3.27	2.90
8		0.18	2.91	3.09
9		0.13	2.35	2.57
10		0.14	2.25	3.53
Average		**0.14**	**2.42**	**3.24**

6.4 Discussion

In the conventional VQ-VAE, the existing research adopted powerful autoregressive deep learning [91, 92] as a decoder network of VQ-VAE for superior reconstruction performance for image and audio data. However, this method required massive computational complexity. In [93], the Multi-Layer Perceptron (MLP) architecture in the network was designed to complete the z-latent dimensionality reduction and flexibility. It can manage the size without using CNN as an encoder coding layer to expand the input data into huge dimensions. The reconstructed output was better than the conventional VQ-VAE in terms of Log Spectral Distortion to quantize the spectral envelope parameter of the WORLD vocoder. In [57], another method to enhance the reconstruction performance was presented. The training technique was based on the Expectation-Maximization (EM) algorithm and achieved better reconstruction performance than the traditional training technique of VQ-VAE on typical encoder and decoder networks. The EM algorithm was applied to the loss criteria to update the embedding space to improve the output reconstruction performance of VQ-VAE.

The minimization of the KL divergence is introduced in the VQ-VAE in low-dimensional representation (z-latent) that is not directly observed but inferred through a mathematical model and stored helpful information from the input data. The GANs do not work with any explicit density estimation like VAE or VQ-VAE, which learn the likelihood distribution through loss function. The GAN finds Nash equilibrium between the generator and discriminator networks using universal function approximators such as neural networks, which explicitly learn the likelihood distribution. A generator learns to capture the data distribution, and a discriminator estimates the probability that a sample came from the data distribution rather than the model distribution.

Reference [18] shows the GAN performance in VAE. The decoder is assumed to be the generator cooperating with the discriminator to find the Nash equilibrium by discriminating whether the reconstructed output is different from the input. This method, called VAEGAN, is our basic idea for developing models. In the experiments, VQ-VAE and GAN adversarial training techniques were investigated in several assumed generator parts in VQ-VAE for spectral envelope parameter quantization. The first proposed model was VAEGAN implemented in VQ-VAE. The discriminator network received the raw SP vector and quantized SP vector to distinguish the difference and return adversarial loss to update the decoder network

parameters in the training process. The second proposed VQ-VAE-EMGAN replaced the adversarial discriminator loss with the mean square error in the VQ loss term to update the embedding space parameters in VQ-VAE. The discriminator that distinguishes between the z-latent and the quantized z-latent is the same or not by Nash equilibrium without sampling the data from the Gaussian distribution like VAEGAN implemented in VQ-VAE. The proposed VQ-VAE-EMDEC model's third model combines the VAEGAN implemented in VQ-VAE and the VQ-VAE-EMGAN together. The discriminator network distinguished between the raw and the quantized SP vectors. It returned the adversarial loss to update the decoder network parameters and the embedding space parameters.

In Table 6.4, the experiments evaluated sixteen models based on deep learning on varied bitrates from four different vector quantization methods: VQ-VAE, VAEGAN implemented in VQ-VAE, VQ-VAE-EMGAN, and VQ-VAE-EMDEC. The VQ-VAE-EMGAN and VQ-VAE-EMDEC models, which include adversarial training on the embedding space, gave better performance by reducing the LSD and L2 and increasing the PESQ score compared to the VQ-VAE. In Figure 6.19, the proposed techniques, the VQ-VAE-EMGAN and VQ-VAE-EMDEC, showed clearer quantized z-latents compared to the VQ-VAE and the VAEGAN implemented in VQ-VAE. On the other hand, the VAEGAN implemented in VQ-VAE model which had the adversarial loss updating only at the decoder network could not operate well at the lower bitrate operation. But at the highest bitrates operation, it increased the performance, as well as the VQ-VAE and VQ-VAE-EMGAN and VQ-VAE-EMDEC.

The results in Table 6.5 represented the effect of each initialization method in terms of the model parameter initialization effect. The model parameters drawn from the normal distribution obtained the best results in terms of performance because the drawn samples had values nearly zeros, based on the mean of normal distribution. On the other hand, the drawn sample values from uniform distribution were too big based on uniform distribution, making it difficult for model weight parameters to converge the loss function. The other investigation presented in Table 6.6 attempted to train the model parameter with normal distribution initialized ten times. The results indicated that each time of parameter initialization also affects the model performance, but the results of each attempt were not significantly different after the finished model training.

6.5 Conclusion

In conclusion, this chapter provides the following contributions:

- We proposed the three models: VAEGAN implemented in VQ-VAE, VQ-VAE-EMGAN, and VQ-VAE-EMDEC as the improved version of VQ-VAE. They combined deep learning adversarial training techniques to increase the VQ-VAE reconstruction performance of the spectral envelope quantization by the well-known GAN technique.

- In the experiment, we designed sixteen spectral envelope parameter quantizers applied to the WORLD vocoder to extract the spectral envelope parameter at 16 kHz sampling frequency speech. The quantization performance in four target bitrate operations varied from low to high bitrates was evaluated. The results showed that the proposed VQ-VAE-EMDEC could reduce the average LSD by around 0.98 points in dB, the average L2 z-latent error by around 0.11, and in terms of reconstructed speech waveform, the proposed method also improved the PESQ by around 0.32, compared to the VQ-VAE.

- The model initialization methods affected the model performance after the finished training process, and every single initialized model parameter also affected a fewer to performance of the model after the finished training.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Through this thesis, we examined the spectral envelope quantization based on deep learning to clear the deep learning performance compared to the conventional quantization techniques. In addition, we implemented five contribution models, which investigate deep learning techniques for spectral envelope quantization. All implementations utilized the WORLD vocoder with high performance to estimate the spectral envelope parameters of the speech data. The datasets used in this dissertation were the VCTK corpus at a sampling rate of 48 kHz and the LibriSpeech corpus at a sampling rate of 16 kHz for investigating the deep learning performance of speech spectral envelope quantization.

The first study of the dissertation was presented in chapter 4; the objective is to study the effect of deep learning architecture on VQ based on deep learning. The conventional VQ technique (K-means) was constructed to quantize the spectral envelope in four different target bitrates. They were compared to the standard VQ-VAE in which the architecture was constructed from the Convolutional Neural Networks and the Multi-layer Perceptron architecture as the Multi-layer Perceptron Vector Quantized Variational AutoEncoder (MLP-VQ-VAE).

The second study presented in chapter 5 aims to confirm the effectiveness of the techniques utilized in the conventional VQ, the Sub-band VQ, and the predictive VQ. We experimented with improving the reconstruction performance when applying those techniques in VQ-based learning. The conventional VQ was constructed to quantize the spectral envelope in four target bitrates compared to the proposed sub-band Vector Quantized Variational AutoEncoder (sub-band VQ-VAE) and the Predictive Vector Quantized Variational AutoEncoder (Predictive VQ-VAE).

The last study in the dissertation presented in chapter 6 was the advanced deep learning training techniques in VQ-VAE, the combination between VQ-VAE and the Generative Adversarial Network (GAN) designed to work

together in the spectral envelope quantization in four different target bitrates compared to the conventional VQ, and VQ-VAE. The experimental results showed the effectiveness of the adversarial loss update on the whole networks of VQ-VAE and only the embedding space of quantization in the VQ-VAE.

In summary, this dissertation provides the following contribution.

1. The study demonstrated that the results in chapter 4 showed that the vector quantization based on deep learning reconstruction performance is better than the conventional vector quantization technique. The followings support this contribution:

    1-I. The study showed that deep learning methods effectively quantize the speech spectral envelope parameters at the high-quality raw speech waveform at 48 kHz called Vector Quantized Variational AutoEncoder by utilizing the WORLD vocoder to represent the estimated high-quality spectral envelope.

    1-II. The study showed that the Vector Quantized Variational AutoEncoder outperformed the conventional Vector Quantization technique (K-means) in terms of reconstruction performance at the same various bitrates for spectral envelope quantization.

    1-III. The study implemented the proposed Multi-Layer Perceptron Vector Quantized Variational AutoEncoder (MLP-VQ-VAE) to quantize the spectral envelope parameters of the high-quality 48kHz WORLD vocoder. As a result, the codebook size of the representation vectors of the MLP-VQ-VAE was around 1.6 times smaller than that of the conventional vector quantization. Moreover, the embedding space and z-latent sizes were around 21 times smaller than the conventional VQ-VAE.

    1-IV. The study showed that the MLP-VQ-VAE achieved better reconstruction performance in terms of Log Spectral Distortion (LSD), and the LSD was reduced by around 1.1 points in dB compared with the conventional vector quantization and by around 2.5 dB compared to the VQ-VAE.

2. The study demonstrated that the results in chapter 5 of the vector quantization based on deep learning reconstruction achieved better

performance by applying the advanced vector quantization techniques. The followings support this contribution:

2-I. We proposed the Sub-band VQ-VAE, which is a combination of a sub-band vector quantization technique and the VQ-VAE, to quantize the spectral envelope parameters of the high-quality 48kHz WORLD vocoder. This model focused on a specific frequency sub-band by assigning more quantization bits and leaving unnecessary frequency sub-band with fewer bits for the bit-allocation.

2-II. The study showed that the proposed Sub-band VQ-VAE performed well in quantizing the spectral envelope parameter of the high-quality WORLD vocoder that operates at 48kHz raw speech waveform. The LSD results in the four various bitrates show that the sub-band VQ-VAE had lower average LSD values than the VQ-VAE, around 0.93 points in dB.

2-III. The study showed that the disadvantage of the proposed Sub-band VQ-VAE needed more substantial embedding space than the conventional VQ-VAE, around 2.17 times. The effective representation of the codebook is a future problem.

2-IV. We proposed the Predictive VQ-VAE, a combination of a predictive vector quantization technique and the VQ-VAE, to quantize the spectral envelope parameters of the high-quality 48kHz WORLD vocoder. The predictive quantization technique utilizes the previous data to produce the current data.

2-V. The study showed that the proposed Predictive VQ-VAE had a lower distortion in LSD for four target bitrates than the conventional VQ-VAE. However, the model complexity increased a lot because the Predictive VQ-VAE required the encoder and decoder predictor networks. Nevertheless, the LSD results showed that the average LDS from four operation bitrates of the Predictive VQ-VAE was lower than the VQ-VAE by around 2.58 points in dB.

3. The study demonstrated the results in chapter 6 of the vector quantization based on deep learning reconstruction performance achieved better performance by applying the advanced deep learning techniques. The following findings support this contribution:

3-I. We proposed three improved VQ-VAE training algorithms: the VAEGAN implemented in VQ-VAE, the VQ-VAE-EMGAN, and the VQ-VAE EMDEC, which implemented advanced deep learning based on the Generative Adversarial Networks (GAN) technique. The introduction of the GAN techniques to update the specific network parameters of the VQ-VAE improved the reconstruction performance of the spectral envelope.

3-II. The study presented experiments of designing four spectral envelope parameter quantizers applied to the WORLD vocoder to extract the spectral envelope parameter at 16 kHz. The quantization performance was evaluated in four target bitrate operations varied from low to high bitrates. The results showed that the proposed VQ-VAE-EMDEC reduced the average LSD by around 0.98 points in dB, the average L2 z-latent error by around 0.11, and the proposed method also increased the PESQ by around 0.32, compared to the VQ-VAE.

3-III. The model initialization methods affected the model performance, and every single initialized model parameter also affected a little bit to performance of the model.

7.2 Future Work

In addition to this dissertation, several improvements can provide a better quantization performance. First, the existing studies in this dissertation utilized the VQ-VAE that the intermediate representation by the vector quantization technique, to force continuous z-latent to be discrete z-latent. The encoder network should produce the discrete z-latent directly, and the decoder network utilizes the discrete z-latent to reproduce the input of the encoder network. With this approach, the reconstruction loss term reduces only to optimize the difference between input and output error by the optimization tool without designing a precise quantizer for the z-latents. Of course, although the encoder and decoder training procedures might be complex and carefully trained, this end-to-end approach will be an important problem for future deep learning vocoder development.

Appendix A

List of Publications and Awards

A.1 International journal paper (peer-reviewed)

[J.1] Srikotr, T., & Mano, K. (2021). Vector Quantization of Speech Spectrum Based on the VQ-VAE Embedding Space Learning by GAN Technique. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E105-A, No.4, pp.647-654, April. 2022.

A.2 International conference papers (peer-reviewed)

[C.1] Srikotr, T., & Mano, K. (2020, January). The multilayer perceptron vector quantized variational autoencoder for spectral envelope quantization. In 2020 IEEE International Conference on Consumer Electronics (ICCE) (pp. 1-6). IEEE.

[C.2] Srikotr, T., & Mano, K. (2019, October). Sub-band Vector Quantized Variational AutoEncoder for Spectral Envelope Quantization. In TENCON 2019-2019 IEEE Region 10 Conference (TENCON) (pp. 296-300). IEEE.

[C.3] Srikotr, T., & Mano, K. (2020, January). Predictive vector quantized variational autoencoder for spectral envelope quantization. In 2020 International Conference on Electronics, Information, and Communication (ICEIC) (pp. 1-4). IEEE.

A.3 International conference paper (non-reviewed)

[C.4] Srikotr, T., & Mano, K. (2019, March). World Vocoder Quantization Based on Adversarial AutoEncoder. In 2019 13th SEATUC Symposium, IS03-23, (pp.406). SEATUC.

A.4 Awards and Scholarships

[A.1] Hybrid Twinning Program Scholarship (HBT), October 2017, Shibaura Institute of Technology.

[A.2] The IEEE CE East Joint Japan Chapter, ICCE Young Scientist Paper Award, January 2020, The IEEE CE East Joint Japan Chapter.

Bibliography

[1] Oppenheim, A. V. (1978). Applications of digital signal processing. Englewood Cliffs.

[2] Proakis, J. G., & Manolakis, D. G. (1992). Digital signal processing. MPC, New York.

[3] Rabiner, L. R., & Gold, B. (1975). Theory and application of digital signal processing. Englewood Cliffs: Prentice-Hall.

[4] Gersho, A. (1994). Advances in speech and audio compression. Proceedings of the IEEE, 82(6), 900-918.

[5] Wong, D., Juang, B., & Cheng, D. (1983, April). Very low data rate speech compression with LPC vector and matrix quantization. In ICASSP'83. IEEE International Conference on Acoustics, Speech, and Signal Processing (Vol. 8, pp. 65-68). IEEE.

[6] Gibson, J. D. (2016). Speech compression. Information, 7(2), 32.

[7] Najih, A. M. M., Ramli, A. R., Ibrahim, A., & Syed, A. R. (2003, August). Comparing speech compression using wavelets with other speech compression schemes. In Proceedings. Student Conference on Research and Development, 2003. SCORED 2003. (pp. 55-58). IEEE.

[8] Anderson, J. B. (2006). Digital transmission engineering (Vol. 12). John Wiley & Sons.

[9] Smith, D. R. (2012). Digital transmission systems. Springer science & business media.

[10] Benedetto, S., & Biglieri, E. (1999). Principles of digital transmission: with wireless applications. Springer Science & Business Media.

[11] Amoroso, F. (1980). The bandwidth of digital data signal. IEEE Communications magazine, 18(6), 13-24.

[12] De Silva, P. Y., & Ganegoda, G. U. (2016). New trends of digital data storage in DNA. BioMed research international, 2016.

[13] Barrekette, E. S. (1974). Trends in storage of digital data. Applied optics, 13(4), 749-754.

[14] Akram, F., ul Haq, I., Ali, H., & Laghari, A. T. (2018). Trends to store digital data in DNA: an overview. Molecular biology reports, 45(5), 1479-1490.

[15] Campanella, S. (1958). A survey of speech bandwidth compression techniques. IRE Transactions on Audio, (5), 104-116.

[16] Shukla, U. P., Patel, N. B., & Joshi, A. M. (2013, March). A survey on recent advances in speech compressive sensing. In 2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s) (pp. 276-280). IEEE.

[17] Kavitha, P. (2016). A survey on lossless and lossy data compression methods. International Journal of Computer Science & Engineering Technology, 7(03), 110-114.

[18] Black, H. S., & Edson, J. O. (1947). Pulse code modulation. Transactions of the American Institute of Electrical Engineers, 66(1), 895-899.

[19] Recommendation, C. C. I. T. T. (1988). Pulse code modulation (PCM) of voice frequencies. In ITU.

[20] Lipshitz, S. P., & Vanderkooy, J. (2004). Pulse-Code Modulation--An Overview. Journal of the Audio Engineering Society, 52(3), 200-215.

[21] Aoki, N. (2008, August). A technique of lossless steganography for G. 711 telephony speech. In 2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (pp. 608-611). IEEE.

[22] Daengsi, T., Wutiwiwatchai, C., Preechayasomboon, A., & Sukparungsee, S. (2012, January). A study of VoIP quality evaluation: User perception of voice quality from G. 729, G. 711 and G. 722. In 2012 IEEE Consumer Communications and Networking Conference (CCNC) (pp. 342-345). IEEE.

[23] Harada, N., Kamamoto, Y., Moriya, T., Hiwasaki, Y., Ramalho, M. A., Netsch, L., Strachurski, J., Miao L., Taddei, H., & Qi, F. (2010, March). Emerging ITU-T standard G. 711.0—lossless compression of G. 711 pulse code modulation. In 2010 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 4658-4661). IEEE.

[24] Schroeder, M. R. (1966). Vocoders: Analysis and synthesis of speech. Proceedings of the IEEE, 54(5), 720-734.

[25] Govalkar, P., Fischer, J., Zalkow, F., & Dittmar, C. (2019, September). A comparison of recent neural vocoders for speech signal reconstruction. In Proc. 10th ISCA Speech Synthesis Workshop (pp. 7-12).

[26] Flanagan, J. L. (1980). Direct Digital‐to‐Analog Conversion of Acoustic Signals. Bell System Technical Journal, 59(9), 1693-1719.

[27] Jung, J. W., & Hawksford, M. J. (2004). An oversampled digital PWM linearization technique for digital-to-analog conversion. IEEE Transactions on Circuits and Systems I: Regular Papers, 51(9), 1781-1789.

[28] Wong, D., Juang, B. H., & Gray, A. (1982). An 800 bit/s vector quantization LPC vocoder. IEEE Transactions on Acoustics, Speech, and Signal Processing, 30(5), 770-780.

[29] Kaltenmeier, A. (1983, April). Implementation of various LPC algorithms using commercial digital signal processors. In ICASSP'83. IEEE International Conference on Acoustics, Speech, and Signal Processing (Vol. 8, pp. 487-490). IEEE.

[30] Macker, J. P., & Adamson, R. B. (1994). A Variable Rate Voice Coder using LPC-10E. NAVAL RESEARCH LAB WASHINGTON DC COMMUNICATION SYSTEMS BRANCH.

[31] Schroeder, M., & Atal, B. S. (1985, April). Code-excited linear prediction (CELP): High-quality speech at very low bit rates. In ICASSP'85. IEEE International Conference on Acoustics, Speech, and Signal Processing (Vol. 10, pp. 937-940). IEEE.

[32] Chen, J. H., Cox, R. V., Lin, Y. C., Jayant, N., & Melchner, M. J. (1992). A low-delay CELP coder for the CCITT 16 kb/s speech coding standard. IEEE Journal on Selected Areas in Communications, 10(5), 830-849.

[33] Nomura, T., Iwadare, M., Serizawa, M., & Ozawa, K. (1998, May). A bitrate and bandwidth scalable CELP coder. In Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181) (Vol. 1, pp. 341-344). IEEE.

[34] Röbel, A., & Rodet, X. (2005, September). Efficient spectral envelope estimation and its application to pitch shifting and envelope preservation. In International Conference on Digital Audio Effects (pp. 30-35).

[35] Paul, D. (1981). The spectral envelope estimation vocoder. IEEE Transactions on Acoustics, Speech, and Signal Processing, 29(4), 786-794.

[36] Van Renesse, R., Birman, K. P., & Maffeis, S. (1996). Horus: A flexible group communication system. Communications of the ACM, 39(4), 76-83.

[37] Carlson, A. B. (2010). Communication system. Tata McGraw-Hill Education.

[38] Garg, V. K., & Wilkes, J. E. (1996). Wireless and personal communications systems. Prentice-Hall, Inc.

[39] Lee, W. C. (2006). Wireless and cellular communications. Mcgraw-Hill Education.
[40] Y. Linde, A. Buzo, and R. Gray, “An algorithm for vector quantizer design,” IEEE Trans. Communications, vol. 28, no. 1, pp. 84–95, 1980.
[41] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in Proc. the fifth Berkeley symposium on mathematical statistics and probability. California, USA, 1967, vol. 1, p. 14.
[42] Z. Huang, C. Weng, K. Li, Y.-C. Cheng, and C.-H. Lee, “Deep learning vector quantization for acoustic information retrieval,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP’14), 2014, pp. 1364–1368.
[43] W.B. Jiang, P.L. Liu, and F. Wen, “An improved vector quantization method using deep neural network,” AEU – International Journal of Electronics and Communications, Vol.72, No.1, pp.178–183, 2017.
[44] A. Roy, A. Vaswani, A. Neelakantan, and N. Parmar, “Theory and Experiments on Vector Quantized Autoencoder,” in arXiv Preprint, arXiv:1805.11063, 2018.
[45] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, “Neural discrete representation learning,” in Proc. NIPS, Dec. 2017, pp. 6306–6315.
[46] Holyoak, K. J. (1987). Parallel distributed processing: explorations in the microstructure of cognition. Science, 236, 992-997.
[47] Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008, July). Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th international conference on Machine learning (pp. 1096-1103).
[48] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in arXiv Preprint, arXiv:1312.6114, 2013.
[49] Ng, A. (2011). Sparse autoencoder. CS294A Lecture notes, 72(2011), 1-19.
[50] Kovenko, V., & Bogach, I. (2020). A Comprehensive Study of Autoencoders' Applications Related to Images. In IT&I Workshops (pp. 43-54).
[51] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation,” in Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1.: Foundations, D. E. Rumelhart, J. L. McClelland and the PDP research group, Eds. Cambridge, Mass.: MIT Press, 1986, pp. 318-362.

[52] S. Araki, T. Hayashi, M. Delcroix, M. Fujimoto, K. Takeda and T. Nakatani, "Exploring multi-channel features for denoising-autoencoder-based speech enhancement," 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, QLD, 2015, pp. 116-120.
[53] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, “Speech enhancement based on deep denoising autoencoder,” in Proc. of INTERSPEECH, 2013, pp. 436–440.
[54] T. Ishii, H. Komiyama, T. Shinozaki, Y. Horiuchi, and S. Kuroiwa, “Reverberant speech recognition based on denoising autoencoder,” in Proc. of INTERSPEECH, Lyon, France, 2013, pp. 3512–3516.
[55] X. Feng, Y. Zhang, and J. Grass, "Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition," in Proc. of ICASSP2014, 2014, pp. 1778-1782.
[56] A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow, “Adversarial autoencoders,” in arXiv preprint arXiv:1511.05644, 2015.
[57] I. Goodfellow, Q. Le, A. Saxe, and A. Ng, “Measuring Invariances in Deep Networks,” in Proc. Neural Information and Processing System, pp. 646-654, 2009.
[58] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, W. W. Cohen, A. McCallum, and S. T. Roweis, Eds., “Extracting and composing robust features with denoising autoencoders,” in Proc. 25th Int. Conf. Mach. Learn. (ICML’08), 2008, pp. 1096–1103.
[59] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, “Contractive Auto-Encoders: Explicit Invariance during Feature Extraction,” in Proc. Int’l Conf. Machine Learning, 2011.
[60] A. Komatsuzaki, “Extractive Summary as Discrete Latent Variables,” in arXiv Preprint, arXiv:1811.05542, 2018.
[61] T. Wu, W. Zhao, E. Keefer, and Z. Yang, “Deep Compressive Autoencoder for Action Potential Compression in Large-Scale Neural Recording,” Journal of neural engineering, vol. 15, no. 6, 2018.
[62] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord, “Unsupervised speech representation learning using WaveNet autoencoders,” in arXiv Preprint, arXiv:1901.08810, 2019.
[63] G. E. Henter, J. L.-Trueba, X. Wang, and J. Yamagishi, “Deep Encoder-Decoder Models for Unsupervised Learning of Controllable Speech Synthesis,” in arXiv Preprint, arXiv:1807.11470, 2018.
[64] N. Weber, L. Shekhar, N. Balasubramanian, and N. Chambers, “Hierarchical Quantized Representations for Script Generation,” in arXiv Preprint, arXiv:1808.09542, 2018.

[65] T. Zhao, K. lee, and M. Eskenazi, “Unsupervised Discrete Sentence Representation Learning for Interpretable Neural Dialog Generation,” in arXiv Preprint, arXiv:1804.08069, 2018.
[66] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “WaveNet: A Generative Model for Raw Audio: A Generative Model for Raw Audio,” in arXiv preprint arXiv:1609.03499, 2016.
[67] W. Ping, K. Peng, and J. Chen, “ClariNet: Parallel Wave Generation in End-to-End Text-to-speech,” in arXiv Preprint, arXiv:1807.07281, 2018.
[68] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in Proc. Advances Neural Information Processing Systems, 2014, pp. 2672–2680.
[69] X. Mao, Q. Li, H. Xie, RYK. Lau, and Z Wang, "Least squares generative adversarial networks," Proc. IEEE International Conference on Computer Vision, pp. 2794-2802, 2017.
[70] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks.," Proc. International Conference on Machine Learning, pp. 214-223, 2017.
[71] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A.C. Courville, "Improved training of wasserstein gans," Proc. Advances in Neural Information Processing Systems, pp. 5767-5777, 2017.
[72] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," Proc. International Conference on Machine Learning, pp. 7354-7363, 2019.
[73] A. B. L. Larsen, S. K. Sonderby, H. Larochelle, and O. Winther. (2015). "Autoencoding beyond pixels using a learned similarity metric," Proc. International Conference on Machine Learning, New York, USA, 2016.
[74] P. Lyu, X. Bai, C. Yao, Z. Zhu, T. Huang, and W. Liu, "Auto-encoder guided GAN for Chinese calligraphy Synthesis," Proc. Int. Conf. Document Anal. Recognit, pp. 1095-1100, 2017.
[75] M. Morise, F. Yokomori, and K. Ozawa, “WORLD: A vocoder-based high-quality speech synthesis system for real-time applications,” IEICE Trans. Inf. Syst., vol. 99, no. 7, pp. 1877–1884, 2016.
[76] R. E. Crochiere, S. A. Webber, and J. L. Flanagan, “Digital coding of speech in Sub-bands,” Bel/ Syst. Tech. J., vol. 55, pp. 1069-1085.

[77] P. H. Westerink, D. E. Boekee, J. Biemond, and J. W. Woods, “Sub-bands coding of images using vector quantization,” IEEE Trans. Commun., vol. 36, no. 6, pp. 713–719, 1988.
[78] Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: An ASR corpus based on public domain audio books," Proc. IEEE ICASSP, 2015.
[79] Cuperman, V. (1982). Adaptive differential vector coding of speech. In Proc. Globecom (pp. 1092-1096).
[80] Cuperman, V., & Gersho, A. (1985). Vector predictive coding of speech at 16 kbits/s. IEEE Transactions on Communications, 33(7), 685-696.Cuperman, V. (1982).
[81] Loizou, P. C. (2011). Speech quality assessment. In Multimedia analysis, processing and communications (pp. 623-654). Springer, Berlin, Heidelberg.
[82] Dubey, R. K., & Kumar, A. (2015, March). Comparison of subjective and objective speech quality assessment for different degradation/noise conditions. In 2015 International Conference on Signal Processing and Communication (ICSC) (pp. 261-266). IEEE.
[83] DeGroot, Morris H. (1980). Probability and Statistics (2nd ed.). Addison-Wesley.
[84] Berger, James O. (1985). "2.4.2 Certain Standard Loss Functions". Statistical Decision Theory and Bayesian Analysis (2nd ed.). New York: Springer-Verlag. p. 60. ISBN 978-0-387-96098-2. MR 0804611.
[85] Prodeus, A. (2016). On Some Features of Log-Spectral Distortion as Speech Quality Measure. Autom. Softw. Dev. Eng. J, 1.
[86] Prodeus, A., & Kotvytskyi, I. (2017, October). On reliability of log-spectral distortion measure in speech quality estimation. In 2017 IEEE 4th International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD) (pp. 121-124). IEEE.
[87] A. Rix, J.G. Beerends, M. Holier and A. Hekstra, "Perceptual evaluation of speech quality (PESQ) - a new method for speech quality assessment of telephone networks and codecs," Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings, 2001.
[88] O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458.
[89] Popescu, M. C., Balas, V. E., Perescu-Popescu, L., & Mastorakis, N. (2009). Multilayer perceptron and neural networks. WSEAS Transactions on Circuits and Systems, 8(7), 579-588.

[90] C. Veaux, J. Yamagishi, and K. MacDonald. “CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit,” 2016.

[91] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, and A. Graves, "Conditional image generation with PixelCNN decoders," Proc. Advances in Neural Information Processing Systems, pp. 4790-4798, 2016.

[92] C. Gârbacea, A. van den Oord, Y. Li, F. S. Lim, A. Luebs, O. Vinyals, and T.C. Walters, "Low bitrate speech coding with VQ-VAE and a WaveNet decoder," Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 735-739, 2019.

[93] T. Srikotr, and K. Mano, "The multilayer perceptron vector quantized variational autoencoder for spectral envelope quantization," Proc. IEEE International Conference on Consumer Electronics (ICCE), pp. 1-6, 2020.

[94] Robbins, H., & Monro, S. (1951). A stochastic approximation method. The annals of mathematical statistics, 400-407.

[95] Kiefer, J., & Wolfowitz, J. (1952). Stochastic estimation of the maximum of a regression function. The Annals of Mathematical Statistics, 462-466.

[96] N. Qian, “On the momentum term in gradient descent learning algorithms. Neural networks,” 12(1), 145-151, 1999.

[97] Hinton, G., Srivastava, N., & Swersky, K. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. Cited on, 14(8), 2, 2012.
https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf
(accessed 2022-08-18)

[98] Kingma, D. P., & Ba, J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[99] Choi, D., Shallue, C. J., Nado, Z., Lee, J., Maddison, C. J., & Dahl, G. E. (2019). On empirical comparisons of optimizers for deep learning. arXiv preprint arXiv:1910.05446.

****