Deep Emotional Text-to-Speech and Voice Conversion

Zijiang Yang

Dissertation
zur Erlangung des Doktorgrades
Dr.-Ing.

Fakultät für Angewandte Informatik
Institut für Informatik
Universität Augsburg

11.12.2024

Erstgutachter:	Prof. Dr. Björn W. Schuller
Zweitgutachter:	Prof. Dr. Elisabeth André

Tag der müdlichen Prüfung: 30.06.2025

Acknowledgement

In November 2016, I started my PhD journey at University of Passau under the supervision of Prof. Björn W. Schuller. My deepest gratitude goes to Prof. Schuller, a patient, friendly and professional Doktorvater. He inspired my continued pursuit of academic research, equipping me with the skills and experiences that will benefit me well in the future.

I would also like to express my thanks to my wife and my best friend, Meishu Song. Throughout these years, she has been my unwavering support, especially during times of doubt and fear. Her encouragement, quoted in the words, 'Just try this, no matter whether it will fail or not,' raised me up during my darkest moments. Beyond emotional support, Meishu's unlimited imagination and outstanding academic ability have also significantly influenced my career. Finding a life partner like her is very rare, and I am grateful every day for the destiny that brought us together.

To my parents, Xiaofeng Yang and Junqun Guo, thank you for your continuous financial and emotional support since the day I was born. In Chinese culture, it is often difficult and uncommon to express gratitude to one's parents, so I want to take this opportunity to do so in words. Thank you for everything you have given me and everything you have done for me. I hope you take pride in this thesis, though I know you have always been proud of me.

Finally, I dedicate this thesis to my daughter, Yingxi Yang, the most adorable and bravest little girl, the apple of my eye. I will give you everything within my power, and let this thesis be the first of many things I offer you.

Zijiang Yang
Tokyo, Japan
On the night of 19th August, 2024

Abstract

Emotional Speech Synthesis (ESS) is a rapidly evolving field, with significant advancements in both Emotional Text-to-Speech (ETTS) and Emotional Voice Conversion (EVC). These two research areas are integral to the development of ESS, aiming at different application scenarios. This thesis researches into the background, state-of-the-art studies and key concepts of ETTS and EVC, providing a comprehensive analysis of their respective methodologies and implementations.

In the ETTS domain, this work presents the design and experimentation of one neutral TTS system and two distinct ETTS systems. These systems are evaluated on various performance metrics to assess their capability in synthesising speech with emotional expression. The ETTS systems leverage transfer learning, highlighting the effectiveness of enhancing emotional expressivity in synthetic speech.

Conversely, the EVC domain is explored through both frame-to-frame and sequence-to-sequence approaches. Two frame-to-frame EVC systems are implemented, focusing on CycleGAN and VAE-GAN models. These two systems are tested and analysed, including objective and subjective evaluations, to determine their performance in converting neutral speech into emotional speech.

Additionally, in order to optimise the speech quality of the converted speech, a sequence-to-sequence EVC systems are developed first, based on an advanced model architecture called Transformer. The experimental results demonstrate the feasibility; however, the findings also result in the necessity for further optimisation to achieve more natural and high-quality output. Challenges such as training strategy, data augmentation and information disentanglement are addressed, offering insights for improvement.

This thesis concludes by outlining the general challenges in ESS, along with an outlook on future developments. The exploration of non-autoregressive models, flow-based TTS and diffusion-based TTS, as well as the integration of large models, are discussed as promising directions for improving ESS. These insights contribute to the ongoing efforts to bridge the gap between state-of-the-art studies and the ultimate goal of achieving the synthesis of natural emotional speech.

Contents

List of Figures

List of Tables

1

Introduction

In recent decades, computers have become an integral part of daily life in various forms, including desktops, laptops, tablets, smartphones, and wearable smart devices. From the perspective of how computers assist humans, they are embedded in work, study and personal life. Consequently, a specific research field focusing on techniques that bridge humans and computers emerged, known as Human-Computer Interaction (HCI) [1]. This interaction encompasses two directions: humans sending signals to the computer and the computer providing feedback to humans.

Traditional HCI methods involve using a mouse, keyboard and joystick to input information [2], while the computer mainly provides feedback via a monitor, displaying text and images. However, this interaction pattern does not align with the way humans typically communicate. Although textual communication (e.g., writing letters) is common, speaking and conversing are the primary modes of human interaction. Thus, one significant natural interaction pattern is for humans to speak directly to the computer, which then responds to humans using speech [3].

Another research field, Artificial Intelligence (AI), shares a similar goal with HCI: creating human-like machines capable of human abilities and foresight [4]. Therefore, speech-based interaction is not only a key objective of AI research but also benefits from AI's rapid technological advancements. In recent years, neural networks, a crucial AI technology, have made it possible to solve the complex tasks of human speech understanding and generation. As a result, Automatic Speech Recognition (ASR) [5]—where a computer understands speech—and speech synthesis—where a computer generates speech—have gained significant attention in the research community.

However, merely understanding linguistic information and generating plain speech falls short of achieving a truly human-like interaction. Human communication conveys two types of information: explicit linguistic content and implicit information about the speaker, such as emotional expression [6]. Therefore, the ability to recognise emotional expression in speech and generate emotionally expressive speech is crucial for more human-like interaction. Consequently, Speech

Emotion Recognition (SER) [7] and emotional speech synthesis [8] have become prominent areas of research.

Affective computing is a field focused on the recognition, detection, interpretation, analysis, and generation of emotional characteristics by computers [9]. The term 'affective' refers to 'emotional' [10], making SER and emotional speech synthesis subfields of affective computing, benefiting from advancements in this area. Although affective computing research contains text, speech and video, studies on speech remain the mainstream, as spoken language is the most natural and common form of human communication [8]. Moreover, with the progress in computational power and neural network architecture, generative models are now capable of producing nuanced speech, which is advantageous for generating emotional speech. Particularly after the successful real-life applications of speech synthesis, such as chatbots [11], the potential for emotional speech synthesis appears bright and promising, both in academic research and application.

Therefore, this thesis focuses on emotional speech synthesis, including both synthesis and conversion. Each chapter provides a comprehensive description of the studies, including the background introduction, literature review, model architecture, experimental setup, results, and discussion. Neutral text-to-speech is introduced in Chapter 2, and emotional text-to-speech is demonstrated in Chapter 3. Regarding emotional conversion, two different schemes—frame-to-frame and sequence-to-sequence—are discussed in Chapter 4 and Chapter 5, respectively. Finally, the challenges and outlook of emotional speech synthesis research are presented in Chapter 6, along with the conclusion of the thesis.

2

Neutral Text-to-Speech

In addition to perceiving information from humans and the environment, expressing information is another crucial process for computers in HCI, as introduced in Chapter 1. A typical method of communication is through spoken language, which is also an intuitive and preferred interactive method for human beings [12]. For example, a simple spoken-language-based HCI system involves two main procedures: the machine receives and understands speech from humans using an ASR system, and then generates responses and synthesises speech using a speech synthesis system, as introduced in this chapter.

2.1 Text-to-Speech

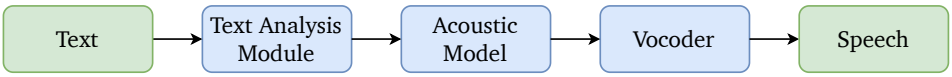
Speech synthesis, also known as Text-to-Speech (TTS), refers to the process and technique of converting written text into speech [13]. Since most research on TTS does not consider emotional expression, it is referred to as neutral TTS in this chapter.

2.1.1 Background

TTS systems have a wide range of real-life applications. Firstly, TTS enhances accessibility for people with visual impairments, enabling them to interact with computers and smart devices [13]. Secondly, TTS provides a means of acquiring information, such as through screen readers, audiobooks and public announcements [12]. Lastly, TTS assists people suffering from aphasia in expressing themselves to others [14].

In addition to these applications, there are numerous potential areas where speech synthesis can offer support or improvement. As a result, many researchers and engineers have sought to synthesise speech using machines. As early as the eighteenth century, a Hungarian scientist built a speaking machine using musical

Figure 2.1: Architecture of the Typical TTS System



instruments [15]. The development of speech synthesis systems has primarily evolved through five stages: Articulatory Synthesis, which simulates human articulators to generate speech [16]; Formant Synthesis, which synthesises speech by manipulating the formant frequencies and spectral properties of human speech [17]; Concatenative Synthesis, which involves concatenating recordings of syllables, words and sentences according to specific rules [18]; Statistical Parametric Speech Synthesis (SPSS), which generates speech from acoustic parameters, such as spectral, pitch and duration parameters [19]; and Neural Synthesis, which utilises Deep Neural Networks (DNNs) to synthesise speech from acoustic features, linguistic features or even directly from input text [20, 21, 22].

Currently, a typical TTS system consists of three components: Text Analysis Module, which extracts linguistic features from the input text; Acoustic Model, which converts linguistic features into acoustic features; and Vocoder, which synthesises the speech waveform using these acoustic features [22]. A typical TTS architecture is shown in Figure 2.1. Additionally, there is research focused on End-to-End models that can synthesise speech directly from input characters [23] or phonemes [24].

2.1.2 Text Analysis Module

The first step in TTS is to extract linguistic information from the input text. Various processing methods are employed to normalise the input text and generate linguistic features. First, the written text must be converted into spoken text to provide pronunciation information, a procedure known as Text Normalisation [22]. In practice, text normalisation can be either rule-based [25] or accomplished using neural networks [26]. Second, for languages such as Chinese and Japanese, correct delimiters between words must be determined through Word Segmentation [27] to prevent misunderstandings. Third, some words have different pronunciations depending on their Part of Speech (POS, also known as word class) in the sentence; this issue can be addressed using POS Tagging [28]. Fourth, to achieve more accurate and natural speech synthesis, prosody information must be predicted, a process called Prosody Prediction [29]. Finally, applying phonemes is generally preferable to using graphemes, as lexicons may not always cover pronunciation in some languages, such as English [22]. Therefore, it is often necessary to convert all words into phonemes, a conversion process known as Grapheme-to-Phoneme [30].

Table 2.1: Examples for Text Analysis Methods

	Input Sequence	**Output Sequence**
Text Normalisation	'200 apples.' 'In 1800.'	'Two hundred apples.' 'In eighteen hundred.'
Word Segmentation	'一点点奶茶'	'一点点 奶茶' '一点 点 奶茶'
POS Tagging	'John keeps ***records***.' 'Alice ***records*** the film.'	'*n. v.* ***n.***' '*n.* ***v.*** *art. n.*'
Prosody Prediction	'I want to ***go***.' 'Do you want to ***go***?'	'I want to ***go***(↘).' 'Do you want to ***go***(↗)?'
Grapheme-to-Phoneme	'Speech' 'Synthesis'	'S P IY CH' 'S IH N TH AH S AH S'

For better understanding, two examples of each text analysis method are provided in Table 2.1.

2.1.3 Acoustic Model

After obtaining linguistic features from text analysis, the next step is to derive acoustic features using these linguistic features (or directly from the input text/phonemes). Over time, three main types of acoustic models have been developed: SPSS-based models [19, 31], Sequence-to-Sequence models [20, 32], and the latest Parallel Generation models, which utilise feed-forward networks [22, 33, 34].

As described in Section 2.1.1, SPSS uses linguistic features to generate acoustic parameters. This modelling can be performed using the more traditional Hidden Markov Model (HMM) [19, 35] or more advanced neural networks [31, 33, 36, 37].

Given the outstanding performance of sequence-to-sequence models in machine translation tasks [38], they were introduced to TTS, treating the task as a form of text-to-speech translation. Sequence-to-sequence learning takes one sequence as input and outputs another sequence. The first attempt at this approach used a Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) model [39]. With the use of an encoder and a decoder, a commonly applied sequence-to-sequence architecture is established [20, 21, 32, 40]. Specifically, an input sequence $\mathbf{X} = (x_1, x_2, x_3, ..., x_T)$ is fed into the encoder to produce the memory $\mathbf{M} = (m_1, m_2, m_3, ..., m_T)$, which represents the hidden states of the input sequence:

$$m_t = enc(x_t, m_{t-1}) \tag{2.1}$$

where the previously generated memory m_{t-1} also contributes in the RNN. To enhance the alignment between the encoder and the decoder, an attention module is typically employed, allowing the model to learn the alignment between the grapheme/phoneme in the input text and its corresponding pronunciation in the output speech [20]. For example, the equation for the context vector c_t in content-based attention is given by [41]:

$$c_t = \sum_{i=0}^{T} \alpha_{ti} m_i \tag{2.2}$$

where α_{ti} is the attention weight, which is computed by *softmax* function:

$$\alpha_{ti} = \frac{exp(e_{ti})}{\sum_{j=0}^{T} exp(e_{tj})} \tag{2.3}$$

where

$$e_{ti} = aln(\hat{y}_{t-1}, m_i) \tag{2.4}$$

is referred to as the alignment model, which evaluates the correlation between the memory around i and the decoder output around t. During the training phase, the decoder utilises the attention vector $\mathbf{C}$, along with the previous element $\hat{y}_{t-1}$ in the target sequence, to predict the output y_t:

$$\hat{y}_t = dec(y_{t-1}, \mathbf{C}) \tag{2.5}$$

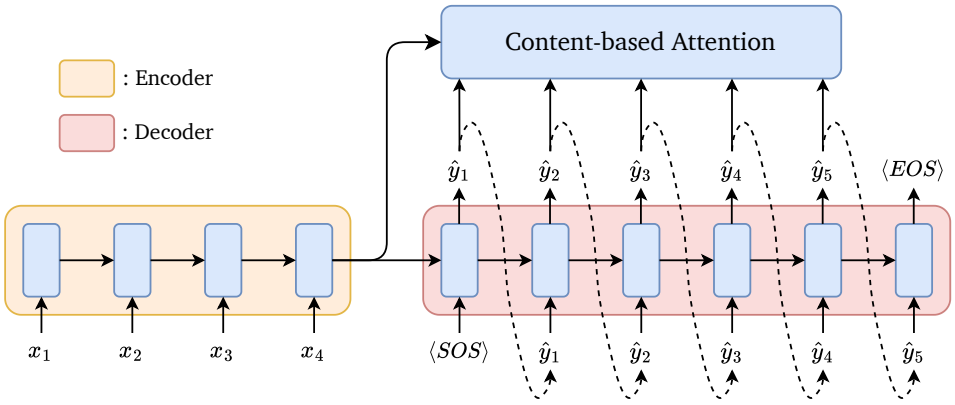
which is also referred to as teacher-forced learning, as it utilises the target (ground truth). However, during the inference phase, where the target is not available, the previous element $\hat{y}_{t-1}$ in the predicted sequence is used:

$$\hat{y}_t = dec(\hat{y}_{t-1}, \mathbf{C}) \tag{2.6}$$

Since the model predicts the output using its own earlier predictions, it is classified as an autoregressive model [42]. The architecture of an RNN-based sequence-to-sequence model with content-based attention is illustrated in Figure 2.2, where $\langle SOS \rangle$ and $\langle EOS \rangle$ represent the Start of the Sequence (SOS) token and End of the Sequence (EOS) token, respectively. The orange section of the figure corresponds to the encoder, while the red section represents the decoder.

Autoregressive RNN models require waiting for the previous prediction before predicting the next one, which results in longer processing times during both the training and inference phases. To address this, Convolutional Neural Networks (CNNs) [32] and self-attention mechanisms [40, 43] were introduced into TTS, successfully accelerating the training phase. However, they still encountered challenges with slow inference [22]. To overcome this, five different models for

Figure 2.2: RNN-based Sequence-to-Sequence Model with Content-based Attention



the parallel generation have been developed: Feed-Forward Transformer [33, 40], Generative Adversarial Networks (GANs) [44], Inverse Autoregressive Flow [45], Generative Flow [46], and Diffusion models [47, 48].

2.1.4 Acoustic Features

For SPSS-based TTS models, the selection of acoustic features and the corresponding vocoder is another key consideration in model design. Two commonly used feature sets include F_0-based features and spectrogram-based features.

The fundamental frequency (F_0) is defined as the smallest inverse of the period of a periodic signal [49] and is often used in combination with other features such as Mel-Cepstral Coefficients (MCEPs, also known as MCCs) [19] and Line Spectral Pairs (LSPs) [37] to synthesise waveforms.

Recently, feature sets like the spectrogram and Mel-spectrogram have been widely adopted in TTS, especially within DNN-based architectures. While the original waveform only conveys information in the time domain, information in the frequency domain is also crucial for speech. Therefore, Fourier Transformation (FT) is employed to convert the waveform into a frequency spectrum:

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(t) \cdot exp(-2\pi i \cdot t\xi) dt \tag{2.7}$$

where ξ represents the frequency, and the complex number $\hat{f}(\xi)$ represents the spectrum of the input waveform $f(t)$ at the frequency ξ. Similarly, the waveform can be reconstructed using Inverse Fourier Transformation (IFT):

$$f(t) = \int_{-\infty}^{\infty} \hat{f}(\xi) \cdot exp(2\pi i \cdot \xi t) d\xi \tag{2.8}$$

However, the discrete Short-Time Fourier Transformation (STFT) [50], which uses a sliding window, is more practical than FT:

$$\hat{f}(\xi, n) = \sum_{k=0}^{l-1} w(k) \cdot f(n \times h + k) \cdot exp(\frac{-2\pi i \cdot \xi k}{l}) \tag{2.9}$$

where n represents the index of the sliding window, w represents the window function, and l and h denote the length and hop length of the window, respectively.

Subsequently, to capture information in both the frequency and time domains:

$$S[f(t)] = \left| \hat{f}(\xi, n) \right|^2 \tag{2.10}$$

where $S[f(t)]$ represents the spectrogram of the input waveform $f(t)$. A notable application of the spectrogram in TTS is Tacotron [20], where the Griffin-Lim algorithm [51] was used to reconstruct the predicted linear-scaled spectrogram back into a waveform.

To enhance the quality of generated speech and avoid characteristic artefacts, Tacotron 2 [21] uses the Mel-spectrogram instead of the linear spectrogram. The Mel-spectrogram is obtained by applying a Mel-scale, a scale designed to reflect the auditory perception of human beings, to the spectrogram.

2.1.5 Vocoder

As discussed in Section 2.1.1, the vocoder's purpose is to convert the output of the acoustic model into a waveform. STRAIGHT [52] and WORLD [53] are two conventional vocoders that use acoustic features as input. Both vocoders employ F_0, MCEPs, and aperiodic features, though they differ in their methods of analysis and extraction. For vocoders that utilise neural networks, CNNs [44, 54, 55, 56, 57, 58, 59] and RNNs [60, 61, 62] are commonly used. Various models, including autoregressive models [54, 61], flow-based models [55, 56], GAN-based models [44, 57], and diffusion-based models [58, 59], have been implemented.

2.2 Application: Tacotron 2 and LPCNet

In this section, a neutral TTS system that implements Tacotron 2 and the LPCNet vocoder will be introduced, analysed, and discussed.

2.2.1 Tacotron 2

Tacotron 2 [21] is a neutral TTS system that improves upon Tacotron [20]. The most noticeable difference between them lies in the acoustic representation and the corresponding vocoder. Tacotron uses a linear-scaled spectrogram and

employs the Griffin-Lim algorithm [51] to reconstruct the waveform, whereas Tacotron 2 utilises a Mel-spectrogram and uses Parallel WaveNet [63] as the vocoder. Additionally, Tacotron 2 replaces the CBHG module—which includes 1D convolution filters, highway networks, and a bidirectional Gated Recurrent Unit (GRU) RNN—with LSTMs and convolutional layers. Moreover, the encoder-decoder attention mechanisms differ: Tacotron 2 deploys location-sensitive attention [64], which accumulates attention weights from previous timesteps, optimising the content-based attention [41] used in Tacotron.

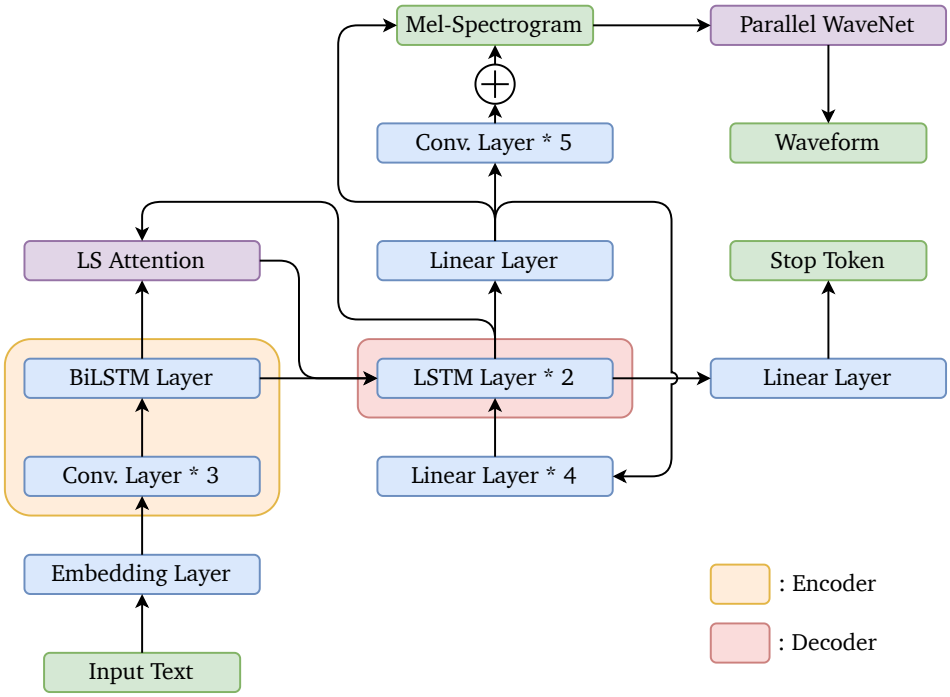
Tacotron 2 simplifies text analysis by directly using characters as input to the encoder. The encoder, which comprises three convolutional layers and one bidirectional LSTM layer, extracts linguistic features. The attention module then uses these hidden features to generate location features. Since Tacotron 2 employs an autoregressive decoder, the previous timestep prediction (with the $\langle GO \rangle$ frame for the first step) is fed back into the decoder after processing through a prenet consisting of four fully connected layers. The previous output features and the attention context vector are then sent to the decoder, which includes two uni-directional LSTM layers. The decoder output and the attention context vector are concatenated and passed to two linear projection layers: one predicts the Mel-spectrogram, and the other predicts the stop token. The predicted Mel-spectrogram is subsequently processed by a postnet, which comprises five convolutional layers with a residual architecture. Finally, Parallel WaveNet [63] converts the post-net output into the waveform [21]. The architecture of Tacotron 2 is illustrated in Figure 2.3.

2.2.2 LPCNet

Although Tacotron 2 improves performance by using Mel-spectrograms compared to Tacotron [21], the Mel-spectrogram's size can still be optimised for real-time processing and reduced storage. For instance, Tacotron 2 employs an 80-dimensional Mel-scale [21], while LPCNet [62] offers a more efficient vocoder by using only 20 dimensions of features, including 18 Bark-frequency Cepstral Coefficients (BFCCs) [65], pitch period, and pitch correlation for speech reconstruction.

LPCNet derives its name from Linear Predictive Coding (LPC), a signal processing technique that predicts the current timestep sample using a linear combination of previous samples [66]. LPCNet improves upon WaveRNN [61], a classic GRU-RNN-based neural vocoder, by introducing linear prediction. The compute prediction block in LPCNet uses the output from the previous 16 timesteps to generate predictions [62]. Additionally, to enhance robustness against noisy input features, LPCNet computes a prediction residual, referred to as 'excitation' which combines the current prediction, previous output and excitation, and adds it to the current output [62]. This approach enables LPCNet to significantly outperform

Figure 2.3: Tacotron 2



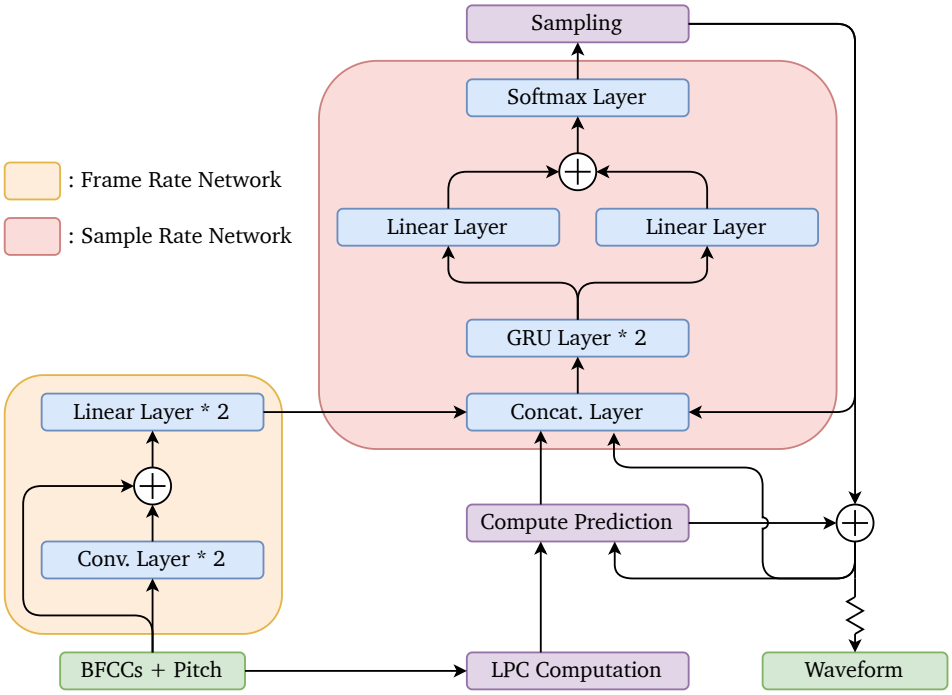
WaveRNN while maintaining the same model size. The architecture of LPCNet is depicted in Figure 2.4.

A straightforward way to combine the advantages of both Tacotron 2 and LPCNet is to use BFCCs and the two pitch parameters as the output features for Tacotron 2 (instead of Mel-spectrograms), and then apply LPCNet to convert the output features into speech. Furthermore, reducing the network size, narrowing the receptive field of convolutional layers and replacing location-sensitive attention with dynamic convolutional attention have been implemented to achieve similar performance to the ground truth [67]. These modifications are primarily aimed at optimising performance for mobile devices, where real-time processing is critical. However, in the context of synthesising emotional speech, where the TTS system functions as a base module rather than the main component, a simple combination of Tacotron 2 and LPCNet is sufficient.

2.2.3 Dataset: Blizzard 2011

Next, it is important to introduce the datasets used for training. Several English datasets have been well-validated and applied in previous studies, including CMU ARCTIC [68], VCTK [69], Blizzard 2011 [70], Blizzard 2013 [71], LJSpeech [72], LibriSpeech [73], and LibriTTS [74].

Figure 2.4: LPCNet



Since 2005, the Blizzard Challenges have aimed to promote the development of corpus-based speech synthesis systems, requiring participants to train TTS models using specified datasets [70, 71]. In 2011, a single-speaker dataset was chosen for training. This dataset was recorded by Nancy Krebs, a female native English speaker with an American accent and also a voice coach [75]. The dataset includes approximately 12,000 utterances, totalling 16.6 hours of recordings at a 16 kHz sampling rate, with corresponding text transcriptions.

2.2.4 Preprocessing

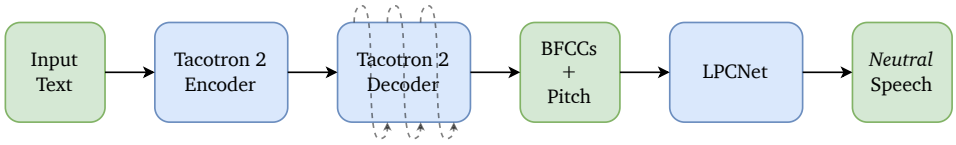
Tacotron 2 uses the characters of the input text as the source sequence [20], making it necessary to modify the input pipeline phoneme-based to prevent mispronunciation. Therefore, a module for grapheme-to-phoneme conversion, known as a phonemiser [76], is essential. In this experiment, a phonemiser based on the CMU Pronouncing Dictionary (version 0.7b) [77] was constructed to generate phonemes from the input text.

The CMU Pronouncing Dictionary (CMU-Dict) includes phonemes for over 134,000 words in North American English, encompassing 39 different phonemes in total [77]. Examples of phonemes for the words ‘speech’ and ‘synthesis’ have been provided in Table 2.1.

Table 2.2: Phonemes in CMU-Dict

AA	*AE*	*AH*	*AO*	*AW*	*AY*	*B*	*CH*	*D*	*DH*	*EH*	*ER*	*EY*
F	*G*	*HH*	*IH*	*IY*	*JH*	*K*	*L*	*M*	*N*	*NG*	*OW*	*OY*
P	*R*	*S*	*SH*	*T*	*TH*	*UH*	*UW*	*V*	*W*	*Y*	*Z*	*ZH*

Figure 2.5: Proposed Neutral TTS System



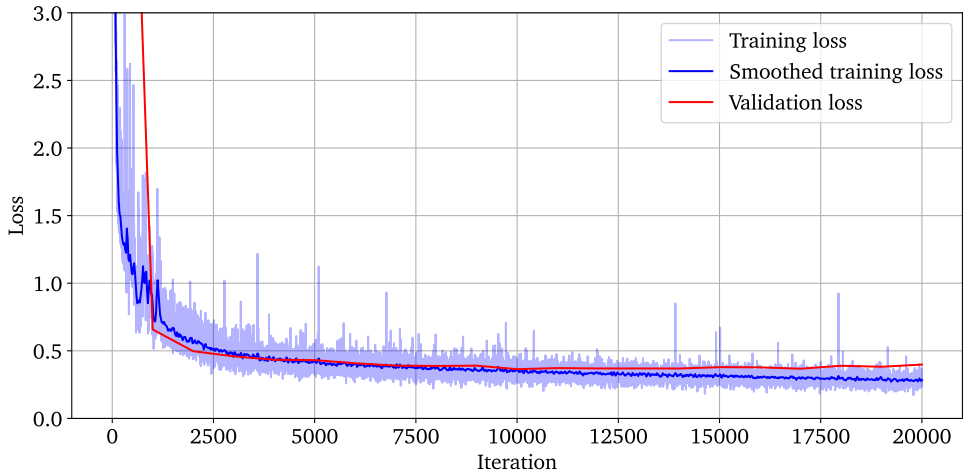
Despite the careful pairing of audio samples with their corresponding prompt texts in the Blizzard 2011 dataset, some audio samples were missing. These incomplete audio-text pairs were removed from the dataset. The remaining prompt texts were normalised according to the text normalisation procedures outlined in Section 2.1.2. A cross-search was then conducted between all the words in Blizzard 2011 and the CMU-Dict, discarding sentences containing words not present in the dictionary, as they could not be phonemised correctly. Consequently, 10,254 usable audio-text pairs remained, which were divided into three subsets for the experiments: 7,191 pairs (70 %) for the training set, 1,541 pairs (15 %) for the validation set, and 1,522 pairs (15 %) for the test set.

2.2.5 Experimental Setup

The cascade architecture combining Tacotron 2 and LPCNet, which represents the simplest scheme for integrating both models, was chosen for implementation. The architecture of this combined system is depicted in Figure 2.5. Since Tacotron 2 originally utilises Mel-spectrograms while the LPCNet employs BFCCs and pitch parameters, any pre-trained Tacotron 2 models could not be used directly. Consequently, a new LPCNet model was trained using the Blizzard 2011 dataset to achieve optimal performance in the final neutral TTS system.

It is important to note that LPCNet's input features comprise a total of 55 dimensions, divided into five parts: 18-dimensional BFCCs, an 18-dimensional zero gap, 2-dimensional pitch features, another 1-dimensional zero gap and 16-dimensional space for iterative prediction. An experimental comparison between generating all 55 dimensions versus only 20 dimensions of features revealed that the 20-dimensional feature set performed better, leading to its selection as the training feature set. Specifically, the generated 20-dimensional features are interpolated with

Figure 2.6: Loss Curves of Tacotron 2 with BFCCs and Pitch Features



gaps according to the structure described, and LPCNet reconstructs the speech using the expanded feature set.

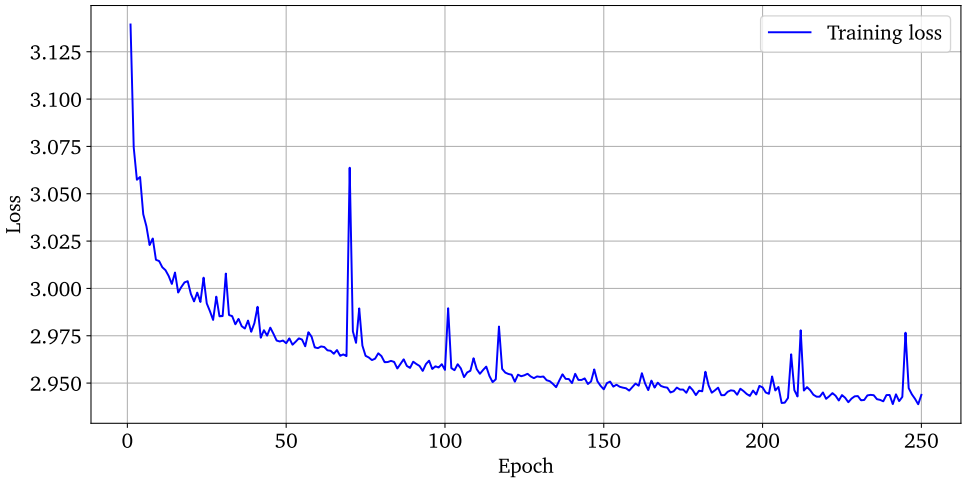
Both Tacotron 2 and LPCNet were trained and fine-tuned on an NVIDIA® TITAN X graphics card. The modified Tacotron 2 model converged to a minimum loss of 0.364 after 90 epochs and 10,000 iterations, as shown in Figure 2.6. To avoid potential overfitting, LPCNet training was early stopped at 155 epochs, achieving a loss of 2.948, as illustrated in Figure 2.7.

2.2.6 Subjective Evaluation

Unlike typical classification or regression tasks, subjective evaluation is more persuasive in speech synthesis tasks because human perception aligns more closely with the ultimate purpose of these tasks than any objective evaluation method [78]. Consequently, a phonetic expert was invited to assess the synthetic speech by comparing 8 pairs of speech samples, each consisting of one natural and one synthetic speech sample expressing the same content. While the synthetic speech demonstrated intonation similar to the natural one, a few segmental distortions on vowels were still observed. To gain a more comprehensive evaluation, the neutral TTS system was assessed subjectively using two metrics: speech emotion classification and Mean Opinion Score (MOS) on naturalness, speech quality, and likeability.

The experiment was designed to evaluate not only the performance of the neutral TTS system but also the emotional voice converter, which is introduced in Chapter 4. A total of 14 participants participated in the subjective evaluation. Eight different sentences were selected, and for each sentence, one ground truth sample (natural recording), one synthetic neutral sample, and eight emotional samples were chosen.

Figure 2.7: Loss Curve of LPCNet in Training Phase



Each participant was asked to listen to a total of 80 ($= 8 \times (1 + 1 + 8)$) samples and determine which emotion they believed the sample expressed. Of the 114 synthetic samples (*8* synthetic samples $\times$ *14* participants), 103 were correctly identified as *Neutral*, resulting in an emotional recognition accuracy of 90.35 %. In comparison, 96 ground truth samples were correctly recognised, achieving an accuracy of 84.21 %. These results indicate that the neutral TTS system, based on Tacotron 2 and LPCNet, was successful and even outperformed natural recordings in emotional expression.

In addition to identifying the perceived emotion, participants were also asked to rate each speech sample on three aspects: naturalness, speech quality, and likeability, using a scale from 1 (worst/least) to 5 (best/most), with an interval of 1. This evaluation method is known as MOS.

The results of the three MOS tests are presented in Figure 2.8. The most significant difference between the ground truth and synthetic speech was in naturalness, where the ground truth achieved a score of 4.7, while the neutral TTS system scored 3.1. This 1.6-point gap suggests that improving the naturalness of the neutral TTS system should be a priority. However, the results in speech quality (0.6-point gap) and likeability (0.8-point gap) were quite promising.

2.2.7 Real-time Performance Evaluation

Furthermore, the real-time performance of the neutral TTS system was assessed using the Real-Time Factor (RTF) as the metric. RTF is defined as the time taken by the system to process one second of speech [53, 67]. The evaluation was conducted on a laptop equipped with an Intel® Core™ i7-8565U CPU, and the results are presented in Tables 2.3 and 2.4. The 'FFmpeg' column in Table 2.3 indicates the

Figure 2.8: Results of the MOS Test on the Proposed Neutral TTS System

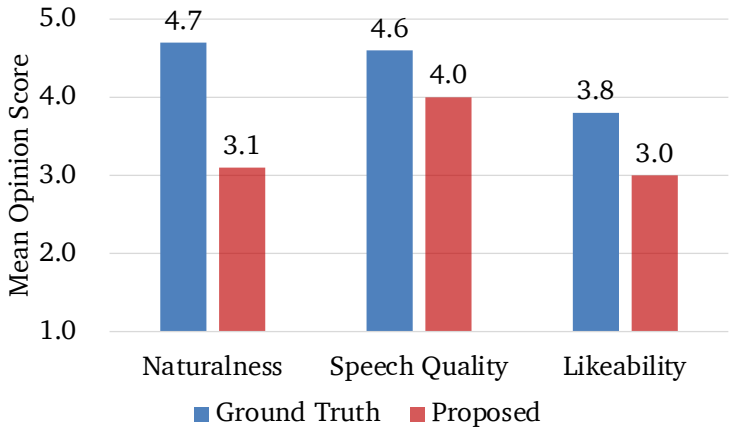


Table 2.3: Results of the Proposed Neutral TTS System on Time Consuming (s)

	Phonemiser	**Tacotron 2**	**LPCNet**	**FFmpeg**	**Σ**
$\boldsymbol{\mu}$	0.1296	2.2133	0.4920	0.0195	2.8546
$\boldsymbol{\sigma}$	0.0059	0.1312	0.0205	0.0008	0.1568
CI (95%)	0.0037	0.0813	0.0127	0.0005	0.0972

RTF performance of the FFmpeg toolkit[1], which was used to convert the 16-bit Pulse-Code Modulation (PCM) files—the output of LPCNet—into waveform files. The result shows that after model loading, the proposed system is able to generate speech in real-time (RTF < 1).

Table 2.4: Results of the Proposed Neutral TTS System on RTF

	Loading	**Tacotron 2**	**LPCNet**
$\boldsymbol{\mu}$	0.9872	0.7747	0.1722
$\boldsymbol{\sigma}$	0.1184	0.0424	0.0059
CI (95%)	0.0734	0.0263	0.0037

[1] `https://www.ffmpeg.org/`

3

Emotional Text-to-Speech

Despite the advancements, neutral TTS systems are not yet perfect in practice. Compared to natural human speech, the generated speech often lacks expressivity, particularly in prosody [79]. Prosody refers to the lexical-independent representation of the acoustic-phonetic properties of speech, including emphasis, pitch accenting, intonational breaks, rhythm and intonation [80]. This limitation has led to increased interest in a new field focused on enhancing the expressivity of TTS systems, known as Expressive/Emotional Speech Synthesis (ESS). In ESS research, models are generally categorised into two primary types based on the input modality they use. This section will focus on the category that generates emotional speech from text input, commonly referred to as Expressive/Emotional Text-to-Speech (ETTS).

3.1 Emotional Text-to-Speech

Some studies on ESS or ETTS, focus solely on enhancing expressivity based on the content of the text, without adhering to a specified expression style. For instance, prosodic information was successfully integrated into TTS systems by extracting prosody embeddings [81]. Similarly, the extraction of global style tokens and their incorporation into Tacotron has also been demonstrated to be effective [82]. Both approaches utilised the audiobook dataset from the Blizzard Challenge 2013 [70], which includes 147 hours of recordings from 49 books, read in an animated and emotive style by a single speaker. Since the expression style in this dataset was not guided by specific instructions, the prosody of the generated speech was determined solely by linguistic information.

Research has shown that prosody affects not only linguistic expression but also emotional expression. In other words, linguistic prosody and affective prosody are closely related [79], and any modification to prosody invariably alters the perceived emotional states [8]. Furthermore, the term 'Expressive' is often considered synonymous with 'Emotional' or 'Affective' in speech synthesis studies [83, 84].

Given these considerations, it is both impractical and unnecessary to draw a clear distinction between expressive speech synthesis and emotional speech synthesis. Therefore, research on either of these topics will be analysed and discussed collectively under the umbrella of ESS or ETTS in this section.

3.1.1 Background

Similar to neutral TTS, ETTS takes the text of a sentence as input and generates speech that incorporates emotional expression. Consequently, ETTS is often considered an extension of TTS [8]. The first attempts at ETTS were made in 1989, as documented in two independent doctoral theses [85, 86]. These early systems modified emotional acoustic correlations in existing speech synthesisers, with the rules for parameter modification determined by experts. Recognition evaluations demonstrated the feasibility of these approaches. Later research explored different ranges of parameters to achieve higher recognition rates, although it still relied on the manipulation of acoustic parameters [87].

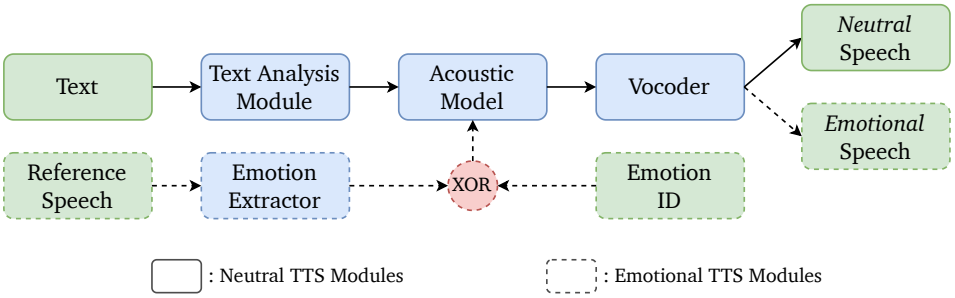
Concatenative synthesis, discussed in Section 2.1.1, was also introduced into ETTS research and gained more attention than the rule-based methods mentioned above [88, 89]. However, this approach faced the same challenge: the need for a large amount of data to select appropriate utterance units for different emotions and speakers, which limited the development of concatenative ETTS [8].

As ETTS research progressed, it began to incorporate SPSS techniques from TTS, where the ETTS system includes an additional input of emotional information within the classic 'text analysis, acoustic model and vocoder' TTS architecture, as described in Section 2.1. Specifically, emotional features are introduced into the system before the acoustic features are passed to the vocoder, which then generates emotional speech. Various network architectures, such as HMM [90], LSTM-RNN [91, 92, 93, 94], CNN [95], and Transformer [96], have been explored in SPSS-based ETTS research. Additionally, emotional information has been leveraged in end-to-end TTS systems [81, 82, 97], and transfer learning has been employed to address the low-resource challenge in ETTS [98]. The distinction between TTS and ETTS is illustrated in Figure 3.1.

3.1.2 Emotional Representation

There are two main approaches to integrating emotional information into a speech synthesis system. The first is reference-based, which utilises an emotional speech sample to specify the target emotional category for the synthetic speech. This approach requires an emotional encoder to extract emotional information from the reference speech [81, 82, 93, 94, 95]. Alternatively, emotional information can be input into the system through emotional vectors, such as one-hot vectors [97, 95] or other representations [92, 91], instead of using reference speech. The reference-based

Figure 3.1: Model Architecture of TTS and ETTS



approach enables finer-grained synthesis compared to using emotional vectors [8]. Moreover, it eliminates the need for the time-consuming and expensive annotation of emotional speech, as reference speech samples can be used instead. However, the choice of emotional representation must align with the specific requirements of different tasks, such as imitating emotion from another speech sample, extracting emotional information from the input text, or determining the required emotional category from an interactive system.

3.1.3 Emotional Datasets

In ETTS research, a variety of datasets have been utilised to understand and replicate human emotions through synthetic voices. As summarised in Table 3.1, it is noteworthy that these datasets are predominantly in Asian languages such as Japanese, Chinese and Korean, with only a few in English. The size of these datasets varies significantly, ranging from approximately $1,800$ samples [98] to 21 hours [97], providing diverse scales for model training and validation.

The emotion categories covered by these datasets also vary significantly. For example, the dataset used in [92] includes a broad range of emotion categories, such as *Calm* and *Insecure*, which are not commonly found in other datasets. In contrast, many datasets primarily focus on more generic emotion categories like *Joyful* and *Sad*.

Moreover, the demographics and number of speakers differ across datasets. While most datasets involve a single female speaker, some, such as those used in [95] and [96], include multiple male and female speakers, with balanced gender representation (5 males and 5 females). This shift indicates a growing interest in capturing a broader spectrum of vocal features and utilising multi-speaker datasets.

It is also worth noting that some studies have specialised focuses. For instance, studies [81] and [82] diverge by targeting expressive TTS using character voices, relying on the same large-scale dataset. In terms of dataset usage, only 10 English speakers were included in the study by [96], while [98] focused on a single female English speaker, highlighting limitations in dataset diversity.

Table 3.1: Information of Emotional Datasets in ETTS Research

Paper	Name	Language	Size	Speaker	Categories
[81][3]	—	English	147 *h*	1 F	Expressive Voices
[82][3]	—	English	147 *h*	1 F	Expressive Voices
[90]	ATR JSD	Japanese	3 018 *samp.*	1 M 1 F	*Reading*, *Joyful*, *Sad*
[91]	—	Chinese	5.5 *h*	1 F	*Neutral*, *Angry*, *Happy*, *Sad*
[92]	—	Japanese	16.95 *h*	1 F	*Neutral*, *Angry*, *Calm*, *Excited*, *Happy*, *Insecure*, *Sad*
[93]	—	Korean	11.7 *h*	1 F	*Neutral*, *Happy*, *Sad*
[94]	—	Korean	3.9 *h*	1 M	*Neutral*, *Angry*, *Happy*, *Sad*
[95]	—	Korean	4 000 *samp.*	5 M 5 F	*Neutral*, *Angry*, *Happy*, *Sad*
[96][1]	ESD [99]	English	13 *h*	5 M 5 F	*Neutral*, *Angry*, *Happy*, *Sad*, *Surprise*
[97]	—	Korean	21 *h*	1 F	*Neutral*, *Angry*, *Fear*, *Happy*, *Sad*, *Surprise*
[98][2]	EmoV-DB [100]	English	∼1 800 *samp.*	1 F	*Neutral*, *Angry*, *Disgust*, *Happy*, *Sleepy*

M: Male
F: Female
samp.: Samples
[1] Only 10 English speakers were included in the experiment.
[2] Only one female English speaker was included in the experiment.
[3] These two papers focused on expressive TTS and used the same dataset.

3.1.4 Evaluation Metrics

The evaluation metrics employed in state-of-the-art ETTS research exhibit both consistency and variety across studies, as detailed in Table 3.2. Objective methods such as Mean Squared Error (MSE), Root Mean Square Error (RMSE), Mel-Cepstral Distortion (MCD) and Voicing Decision Error (VDE) are notably used in works such as [91] and [95]. MSE is frequently utilised not only as a loss function but also as an objective evaluation metric to measure the difference between the synthetic and target speech signals. MSE quantifies the average squared difference between corresponding values in the synthetic sequence $\hat{\mathbf{Y}}$ and the target sequence $\mathbf{Y}$, providing a comprehensive numerical indicator of the model's performance. It is expressed mathematically as:

$$MSE(\mathbf{Y}, \hat{\mathbf{Y}}) = \frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2 \tag{3.1}$$

where N is the length of the synthetic and target sequence.

Similarly, RMSE is essentially the square root of the MSE, offering a more interpretable scale of the errors by converting them back to the original units of the data. RMSE can be computed by using the following equation:

$$RMSE(\mathbf{Y}, \hat{\mathbf{Y}}) = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2} \tag{3.2}$$

MCD is also a widely used objective evaluation metric in TTS and ETTS research for assessing the similarity between synthetic and target speech signals [101]. MCD calculates the distance between Mel-Frequency Cepstral Coefficients (MFCCs), which are compact representations of the spectral envelope, for each frame of speech. The MCD is computed as follows:

$$MCD(\mathbf{M}, \hat{\mathbf{M}}) = \frac{1}{T}\sum_{t=0}^{T-1}\sqrt{\sum_{n=1}^{N}(m_{t,n} - \hat{m}_{t,n})^2} \tag{3.3}$$

where T represents the total number of frames, and N is the number of dimensions of the target MFCCs $\mathbf{M}$ and the synthetic MFCCs $\hat{\mathbf{M}}$. $m_{t,n}$ and $\hat{m}_{t,n}$ denotes the target and synthetic elements at t-th frame and n-th dimension, respectively.

VDE, also known as Voiced/Unvoiced Error Rate, is a metric used to evaluate the accuracy of voicing decisions in speech synthesis and recognition tasks [102]. Specifically, it quantifies the discrepancy between the target and synthetic speech in terms of their voicing decisions across a series of frames. A voicing decision is essentially a binary value indicating whether a particular frame in a speech is voiced

or unvoiced. The equation of VDE between two voice decision sequences $\mathbf{V}$ and $\hat{\mathbf{V}}$ is expressed as:

$$VDE(\mathbf{V}, \hat{\mathbf{V}}) = \frac{\sum_{t=0}^{T-1} 1[v_t \neq \hat{v}_t]}{T} \tag{3.4}$$

where T represents the total number of frames, v_t and $\hat{v}_t$ are the voice decisions of the t-th frame in the target and the synthetic speech, respectively. The value of indicator function $1[\cdot]$ is set to 1 when the equation in the brackets is true, and 0 otherwise. All these four evaluation metrics have the characteristic that lower values indicate greater similarity between the synthetic and the target speech.

Yet, several studies, such as [90], [92] and [93], waive objective metrics altogether in favour of purely subjective methods. Among these, the MOS test stands out as the most widely used, often employed to measure various dimensions such as naturalness, similarity, quality, intensity, and emotional expressivity. As described in Section 2.2.6, MOS, including Degradation MOS, is the most applied subjective evaluation method in state-of-the-art research. However, there are some variations in the classic MOS setting. For example, [91] applied a MOS test on naturalness with a range from 0 to 5, while a range from 1 to 7 was set for the intensity MOS test.

Furthermore, SER is recurrent as a subjective evaluation method in studies such as [90] and [93], although [96] integrates it by using a pre-trained speech emotion classifier.

A few studies apply distinct evaluation strategies; for instance, [92] incorporates Frobenius Distance between the confusion matrix of the evaluated model and the ground truth, and [81] introduces an ABX subjective method to measure the similarity. Thus, while there is a general trend towards subjective methods, especially MOS, the field is marked by a continual search for comprehensive and context-specific evaluation frameworks.

3.2 Application: Transfer Learning

The first ETTS application, which is based on transfer learning, will be introduced in this section. However, before presenting the system, it is important to explain why transfer learning was chosen as the initial approach.

3.2.1 Emotional Speech Datasets

All datasets applied in the state-of-the-art ETTS research are summarised in Table 3.1. While EmoV-DB [100] and ESD [99] are publicly available emotional speech datasets, most other datasets in studies are either not publicly available or

Table 3.2: Evaluation Metrics in ETTS Research

Paper	Objective Methods	Subjective Methods	Comments
[81]	MCD, GPE, VDE, FFE	ABX[1]	[1]'AXY' in paper [1]Range:-3–3
[82]	WER, LDA	PT[2](*spk.*), MOS (*natl.*)	[2]'Side-by-side' in paper
[90]	—	SER, MOS (*simil.*)	—
[91]	MSE (Duration, F_0), MCD, VDE	SER, MOS (*natl.*[3], *int.*[4])	[3]Range: 0–5 [4]Range: 1–7
[92]	—	FD[5], MOS (*int.*, *qual.*)	[5]Involving crowd-sourcing
[93]	—	SER, MOS (*qual.* & *expr.*[6])	[6]Simultaneously
[94]	—	MOS (*qual.*, *like.*)	—
[95]	MCD, BAPD, RMSE (F_0), VDE	MOS (*natl.*), DMOS (*simil.*[7], *simil.*[8])	[7]Speaker [8]Emotion
[96]	WER, SER[9]	MOS (*natl.*, *simil.*)	[9]Pre-trained
[97]	—	—	Online demos
[98]	WER[10]	MOS (*expr.*[11])	[10]Word accuracy [11]Range: 0–5

ABX: Identify X as either A or B
BAPD: Band Aperiodicity Distortion
DMOS: Degradation Mean Opinion Score
FD: Frobenius Distance
FFE: F_0 Frame Error
GPE: Gross Pitch Error
LDA: Linear Discriminative Analysis
MOS: Mean Opinion Score
MSE: Mean Square Error
PT: Preference Test
RMSE: Root Mean Square Error
SER: Speech Emotion Recognition
VDE: Voicing Decision Error
WER: Word Error Rate
expr.: Expressivity
int.: Intensity
like.: Likeability
natl.: Naturalness
qual.: Quality
simil.: Similarity
spk.: Speaker

their names are not disclosed. Consequently, it is essential to review all possible emotional speech datasets.

The landscape of emotional speech and multimodal datasets is diverse in terms of content and methodological design. Table 3.3 provides a comprehensive summary of these datasets, highlighting key dimensions such as language, modalities, size, triggers, number of actors, emotional models and number of annotators.

Regarding language, the datasets cover various languages, including English (IEMOCAP [103], MSP-IMPROV [104], RAVDESS [105], CREMA-D [106]), French (RECOLA [107]), Italian (DEMoS [108]) and German (EMO-DB [109], FAU-AIBO [110]). The EmoV-DB [100] and ESD [99] datasets are bilingual, covering English/French and English/Chinese, respectively, which enhances their utility for cross-lingual studies. Most datasets, such as IEMOCAP, MSP-IMPROV, RECOLA, RAVDESS, and CREMA-D, include both audio and visual components, while DEMoS, EMO-DB, FAU-AIBO, EmoV-DB, and ESD focus solely on audio.

The sizes of datasets vary significantly. For example, IEMOCAP contains 12 hours of data, RECOLA has 9.5 hours, while RAVDESS and CREMA-D are sample-based with 1,440 and 7,442 samples, respectively. FAU-AIBO and DEMoS are larger, with around 9 hours of audio each. ESD is the largest, boasting $35,000$ samples across 2 languages and 20 speakers. The number of actors and their gender distribution are also different. IEMOCAP and EMO-DB each comprise 10 actors (5 male, 5 female), while DEMoS involves 68 actors (45 male, 23 female), providing greater demographic diversity.

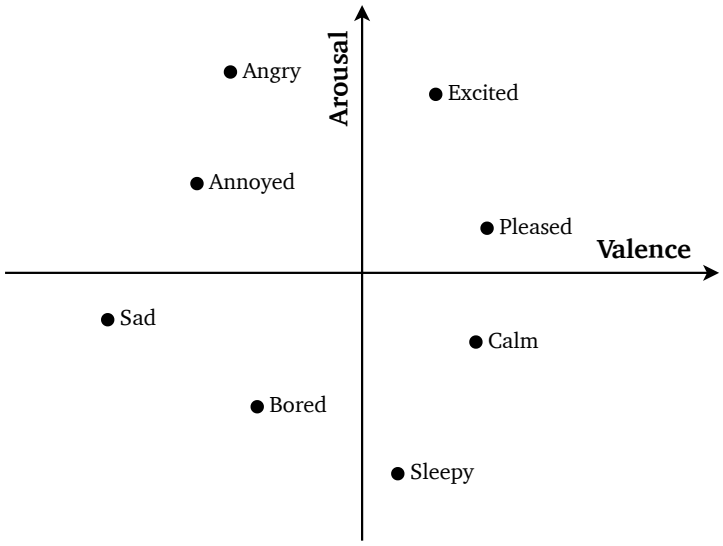
The elicitation method for emotional expression within each dataset is denoted as the emotional trigger, and it manifests in three primary categories: Acted, Re-acted, and Inter-acted, as delineated in prior literature [111]. In the acted paradigm, participants, often professionally trained actors, are instructed to simulate designated emotional states during data recording. The re-acted modality entails exposing participants to specific visual or auditory stimuli—such as scenes from a horror film—and subsequently recording the resultant emotional expressions, for instance, a state of fear. The inter-acted category indicates spontaneous emotional expressions that emerge during participant interactions, such as during an interview session. A review of existing emotional datasets reveals the advantage of the acted elicitation method. However, datasets like RECOLA and FAU-AIBO deviate from this norm by incorporating inter-acted triggers, while the DEMoS dataset uniquely employs the re-acted elicitation approach.

Table 3.3: Information of Reviewed Emotional Speech Datasets

Dataset	**Language**	**Modalities**	**Size**	**Triggers**	**Actors**	**Model**	**Annotators**
IEMOCAP [103]	English	Audio Video	12 *h*	Acted	5 M 5 F	9 Classes V/A/D	2
MSP-IMPROV [104]	English	Audio Video	9 *h*	Acted	6 M 6 F	4 Classes	Crowd-source
RAVDESS [105]	English	Audio Video	1,440 *samp.*	Acted	12 M 12 F	8 Classes	Pre-defined
CREMA-D [106]	English	Audio Video	7,442 *samp.*	Acted	48 M 43 F	6 Classes	Crowd-source
RECOLA [107]	French	Audio Video	9.5 *h*	Inter-acted	19 M 27 F	V/A	6
DEMoS [108]	Italian	Audio	7.8 *h*	Re-acted	45 M 23 F	8 Classes	3 Self-report
EMO-DB [109]	German	Audio	500 *samp.*	Acted	5 M 5 F	7 Classes	20
FAU-AIBO [110]	German	Audio	9 *h*	Inter-acted	21 M 30 F	11 Classes	5
EmoV-DB [100]	English French	Audio	7,590 *samp.*	Acted	3 M 2 F	5 Classes	Pre-defined
ESD [99]	English Chinese	Audio	35,000 *samp.*	Acted	10 M 10 F	5 Classes	Crowd-source

A: Arousal **D**: Dominance **V**: Valence ***samp.***: Samples

Figure 3.2: Schematic Diagram of Valence-Arousal Dimensional Model



Another critical dimension in the configuration of emotional datasets is the emotional model employed. Two predominant frameworks exist for emotion modelling in these datasets: Categorical models and Dimensional models, as introduced in the existing literature [111]. The categorical model posits that a finite set of basic emotional states suffices for the accurate characterisation and representation of emotional expression. For example, the ESD dataset applies this approach by comprising $35,000$ samples across 5 distinct emotion categories, specifically: *Neutral*, *Angry*, *Happy*, *Sad* and *Surprise*.

In contrast to the categorical model, the dimensional model, also known as the continuous model, represents another prevalent methodology for the description of emotional states. This model employs two dimensions to characterise an emotional state, namely Valence—which ranges from negative to positive—and Arousal—which ranges from relaxed to excited [112]. For example, in a dimensional model, the categorical emotion of *Angry* would be situated at a point representing high arousal and low valence, as illustrated in Figure 3.2. An extended version of this model incorporates a third dimension, Dominance [113]. Importantly, categorical and dimensional models are not mutually exclusive and can coexist within the same emotional dataset. As an illustrative case, the IEMOCAP dataset adopts a categorical approach, directing actors to express nine different emotional states such as *Neutral*, *Angry*, *Excited*, *Fear*, *Sad*, *Surprised*, *Frustrated*, *Happy* and *Disappointed*. Concurrently, the dataset also employs a 3-dimensional model for annotation; two evaluators are tasked with rating each sample along the axes of valence, arousal and dominance, using a scale that ranges from 1 (indicating negative, relaxed or weak) to 5 (indicating positive, aroused or strong) [103].

Overall, Table 3.3 provides a structured overview of the state-of-the-art emotional datasets, thereby aiding researchers in making informed choices suitable for their specific research requirements. Notably, a majority of the emotional datasets are suboptimal for ETTS applications due to their design orientation toward perceptual tasks such as ASR and SER. For instance, the IEMOCAP dataset instructs one male and one female actor to recite dialogue transcription while expressing predefined emotional states [103]. As a result, environmental noises and overlapping speech—which substantially influence the quality of the synthetic speech—are inevitable. Likewise, while the CREMA-D dataset comprises a seemingly robust set of $7,442$ speech samples, it suffers from limitations in lexical variability [99, 106]. Specifically, the dataset contains only 12 distinct sentences, each of which is presented in 6 different emotional states by 91 actors. This narrow lexical range restricts its utility for ETTS tasks that require a richer vocabulary. Upon analysing all datasets listed in Table 3.3, it becomes obvious that EmoV-DB and ESD are the only two datasets that meet the criteria requirement for ETTS applications. In contrast to the datasets utilised in the TTS research, the sizes of both EmoV-DB and ESD are comparatively limited in scale. Consequently, transfer learning emerges as an intuitive and preferred approach for addressing this low-resource problem.

3.2.2 Transfer Learning

Transfer learning is a machine learning training technique where a model pre-trained for a particular task is adapted for another, usually related task. Transfer learning involves two distinct elements: a source domain D_s associated with a learning task T_s, and a target domain D_t associated with a learning task T_t. The objective of transfer learning is to enhance the learning performance of a target predictive function $f_t(\cdot)$ within D_t by leveraging the knowledge obtained from D_s and its associated task T_s. Crucially, for transfer learning to be applicable, either D_s and D_t must differ ($D_s \neq D_t$), or T_s and T_t must not be identical ($T_s \neq T_t$) [114]. Traditional supervised learning approaches often require a large amount of labelled data and computational resources for model training. However, transfer learning makes it possible for a model to leverage the pre-existing knowledge gained from a 'Pre-training' task (source domain) to improve its performance on a new but related 'Fine-tuning' task (target domain). Therefore, this approach is particularly beneficial when the target domain has limited labelled data available.

Transfer learning has proven its efficacy across diverse research domains, notably encompassing Natural Language Processing (NLP) [115, 116], Computer Vision (CV) [117, 118], Computer Audition (CA) [119, 120], and Artificial Intelligence Generative Content (AIGC) [121, 122]. Within the realm of affective computing, the integration of transfer learning strategies has enhanced the Unweighted Average Recall (UAR) of a sparse-Autoencoder-based SER model, and this enhancement

has been validated across various emotional speech datasets [123]. Concurrently, for similar SER tasks, Deep Belief Networks (DBNs) have been integrated with transfer learning across five datasets containing three different languages: English, German and Italian [124]. This research highlights how well the suggested SER system works in multi-lingual situations, especially benefiting lesser-known languages that often lack available datasets. Due to its excellent efficacy in affective learning, transfer learning has gained widespread adoption. Consequently, resources like the multi-corpora dataset EmoSet and the pre-trained transfer learning framework EmoNet are readily accessible for academic exploration [125].

While transfer learning offers significant advantages, it is not without any drawbacks. One major concern is the risk of negative transfer, where the model's performance decreases due to irrelevant or misleading information from the source domain. To circumvent this limitation, it is crucial to implement prior studies on the transferability between the source and target domains or tasks [114]. Furthermore, it is worth noting the potential difference in feature spaces between the source and target domains, even though the same spaces are typically observed in the majority of cases [114]. Additionally, the adaptability of the model may require careful fine-tuning to avoid overfitting to the new dataset. In other words, the determination of an optimal termination criterion for fine-tuning needs to be strictly considered.

To summarise, transfer learning provides a robust methodology for improving the performance of models in low-resource machine learning tasks. However, its effective implementation demands careful planning and consideration of various risks and challenges.

3.2.3 Dataset: EmoV-DB

As described in Section 3.2.1, EmoV-DB is one of the qualified datasets for ETTS model training, even though the size of EmoV-DB is very limited. However, transfer learning makes it possible to use a small dataset in the model training. The information of every speaker and the number of samples for every speaker in every emotional state are given in Table 3.4.

The table presented the composition of the EmoV-DB, breaking down the dataset by speaker, language, gender and emotion categories. The emotion categories included are *Neutral*, *Amused*, *Angry*, *Disgust* and *Sleepy*. Four speakers expressed emotions in English, including two male and two female speakers, while one male speaker used French. Each speaker's contributions to the emotion categories are specified, along with the total number of utterances per speaker. Particularly notable is that the speaker *Spk-Sa* has the highest total utterances with $2,454$ across all emotion categories, while the speaker *Spk-Jsh* and *Spk-No* have missing entries for some emotions, indicating a lack of data in those particular categories. Besides, all speakers that incorporate all emotional categories are in English. Therefore, after discarding these two speakers with missing categories, Table 3.5 describes the

Table 3.4: Details of EmoV-DB

Speaker	**Language**	**Gender**	**Emotion Categories**					**Σ**
			neu.	***amu.***	***ang.***	***dis.***	***sle.***	
Spk-Je	English	F	417	222	496	189	466	1,790
Spk-Bea	English	F	357	296	304	333	497	1,787
Spk-Sa	English	M	492	501	468	497	495	2,453
Spk-Jsh	English	M	302	298	—	—	263	863
Spk-No	French	M	317	—	273	—	—	590

neu.: *Neutral* ***amu.***: *Amused* ***ang.***: *Angry* ***dis.***: *Disgust* ***sle.***: *Sleepy*

Table 3.5: Duration Information of the Three Qualified Speakers in EmoV-DB

Speaker	**Gender**	**Emotion Categories**					**Σ**
		neu.	***amu.***	***ang.***	***dis.***	***sle.***	
Spk-Je	F	$31'\,31''$	$18'\,22''$	$43'\,42''$	$15'\,35''$	$37'\,58''$	$2h\,27'\,08''$
Spk-Bea	F	$23'\,23''$	$20'\,29''$	$19'\,03''$	$29'\,19''$	$51'\,13''$	$2h\,23'\,27''$
Spk-Sa	M	$29'\,18''$	$54'\,20''$	$31'\,01''$	$53'\,01''$	$52'\,36''$	$3h\,40'\,16''$

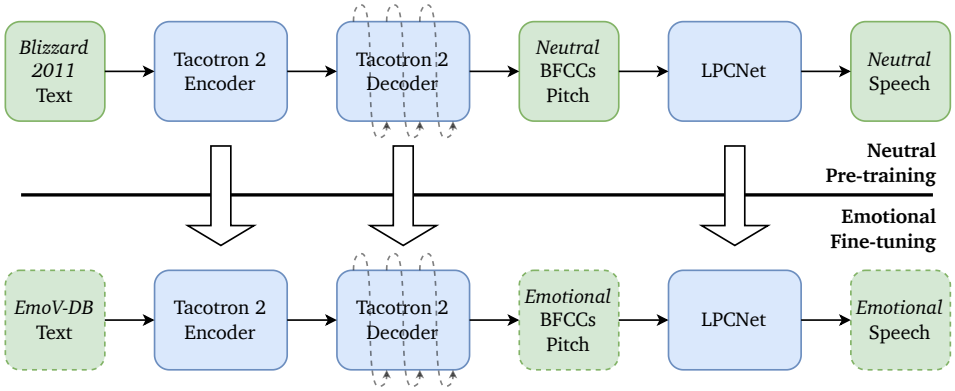
neu.: *Neutral* ***amu.***: *Amused* ***ang.***: *Angry* ***dis.***: *Disgust* ***sle.***: *Sleepy*

EmoV-DB dataset from the perspective of the length of the recordings. This data offers precise insight into the temporal distribution of emotional expressions within the EmoV-DB dataset. Among them, the male speaker, *Spk-Sa*, has the highest total duration of utterances at 3 h 40 m 16 s. Conversely, speaker Spk-Bea has the least total duration of 2 h 23 m 27 s, although closely followed by Spk-Je (2 h 27 m 08 s).

3.2.4 Tacotron 2 with EmoV-DB

The employment of a neutral TTS system as the source task for the transfer learning in divers other speech synthesis tasks is considered highly beneficial due to the obvious similarities among these distinct tasks [121, 126]. Given the limited size of the EmoV-DB, as well as the pre-trained neutral TTS system introduced in Section 2.2, an intuitive effort towards ETTS is the exploitation of transfer learning from the neutral TTS model. Specifically, the emotional speech dataset, EmoV-DB, is utilised for the fine-tuning of the neutral TTS model, which is a Tacotron 2 model

Figure 3.3: Architecture of the Transfer Learning via Tacotron 2 and EmoV-DB



with LPCNet vocoder and initially trained by using the Blizzard 2011 dataset. The architecture of this transfer learning process is shown in Figure 3.3.

Upon careful examination of the datasets presented in Table 3.4 and 3.5, it is obvious that for a transfer learning scheme where the source task involves a TTS model trained on the voice of an actress (Blizzard 2011), *Spk-Bea* appears to be the optimal choice of speaker. The reason for this decision is multifaceted:

- **Gender Relevance**. *Spk-Bea* is one of the female speakers, aligning with the gender-specific requirement of the source task for better knowledge transfer. This alignment is crucial for preserving gender characteristics in the speech synthesis process.

- **Data Volume**. According to both tables, it is shown that *Spk-Bea* contributes a total of $1,866$ emotional samples and the duration is around $2\,h\ 23\,m$, which, while not the largest nor longest in the dataset, presents a substantial volume of data crucial for a robust transfer learning process.

- **Balanced Data Distribution**. A balanced data distribution among different emotions is vital for training a model with a well-rounded understanding and capability of speech synthesis in different emotional states. The data from *Spk-Bea* illustrates a relatively balanced distribution across all five emotion categories, thereby enhancing the general performance of the fine-tuned ETTS system.

Nevertheless, another notable challenge arises relating to the phonemic representation of the text. The CMUDict, utilised in the neutral TTS system for deriving phonemes from the input sentence, incorporates a total of 84 distinct phonemes. However, an absence of 16 phonemes is observed within the text transcription of EmoV-DB, including *AA*, *AE*, *AH*, *AO*, *AW*, *AY*, *EH*, *ER*, *EY*, *IH*,

IY, *OW*, *OY*, *OY0*, *UH*, and *UW*. Furthermore, certain words present in EmoV-DB, such as 'self-esteem' and 'provocateur', are not collected within the CMUDict. While unknown words can be addressed by means of segmentation ('self' and 'esteem') or looking up from alternative dictionaries, no effective measure is available for the missing phonemes. This phonemic problem may not obstruct the training process but could potentially negatively impact the performance of the fine-tuned ETTS model.

A total of four distinct models have been employed, each dedicated to a specific emotional category, whilst maintaining the model architecture of the neutral TTS system. The reason for avoiding the utilisation of an emotional control module is the potential for any newly implemented module to affect the knowledge transfer negatively. Moreover, the fine-tuning was not only applied to Tacotron 2; the vocoder LPCNet was also fine-tuned by using the EmoV-DB dataset to enhance the vocoding performance.

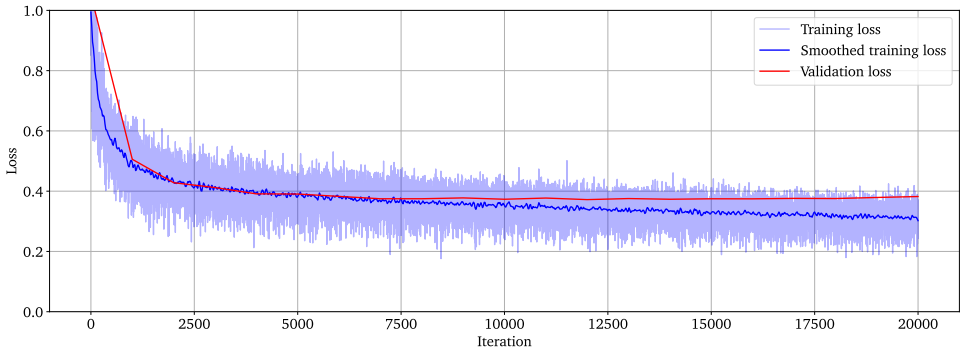
Since the number of samples in each emotional category of the EmoV-DB dataset is limited, the same 10 samples were selected as the validation set for each emotional TTS model during the training phase, allowing more samples to be allocated to the training set. The training results are depicted in Figure 3.4 as loss curves. Four emotional categories—*Amused*, *Angry*, *Disgust* and *Sleepy*—achieved their lowest validation losses of 0.372, 0.357, 0.293 and 0.236 at the iterations of $12,000$, $11,000$, $13,000$ and $16,000$, respectively. Notably, three of these categories showed lower validation losses than the neutral TTS system, which achieved a validation loss of 0.364. However, it is important to remember that the ultimate goal of the ETTS task is human perception, which holds greater significance than numerical metrics, even though these metrics represent the 'distance' between the generated speech and the target speech.

The *Angry* ETTS model demonstrated commendable performance in synthesising angry speech from sentences not present in the training dataset. The effectiveness of this model was further evaluated across various sentences. However, the models corresponding to the other three emotional categories—*Amused*, *Disgust* and *Sleepy*—did not achieve satisfactory results. Despite the clear expression of emotions, the synthetic speech exhibited a significant loss in intelligibility. This suggests that while the output retained the human-like vocal characteristics, the clarity of the content was compromised, rendering the speech incomprehensible.

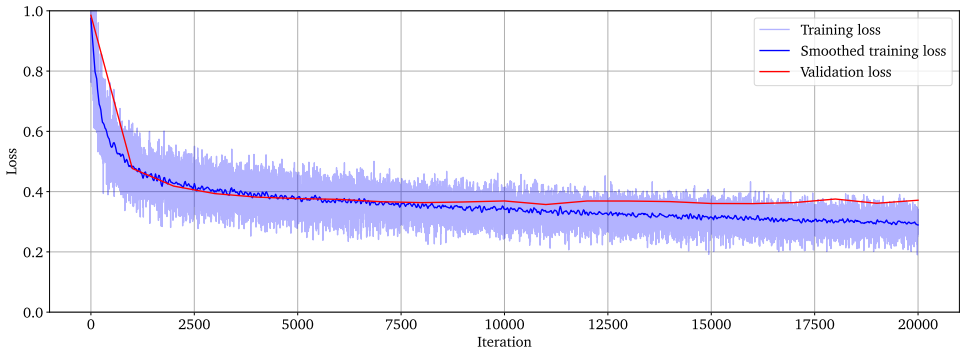
Upon closer examination of both the model and the dataset, the reason behind the *Angry* ETTS model's success becomes evident. In the EmoV-DB dataset, actors and actresses were instructed to express sentences while embodying specific emotional states. To enhance emotional expressivity, performers often incorporated non-verbal utterances—such as laughter, sighs and yawns—particularly in the *Amused*, *Disgust* and *Sleepy* categories. However, these non-verbal utterances were not annotated in the text transcriptions of the EmoV-DB dataset. This oversight caused the model to mistakenly identify non-verbal utterances as phonemes in the

Figure 3.4: Loss Curves of ETTS System of Tacotron 2 with EmoV-DB

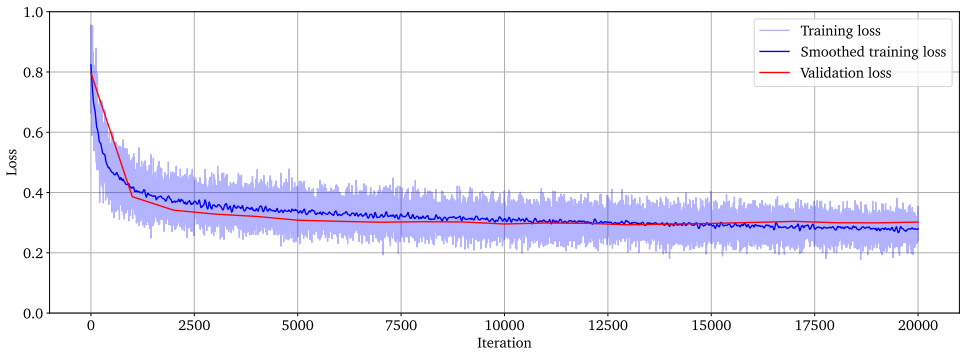
(a) *Amused*



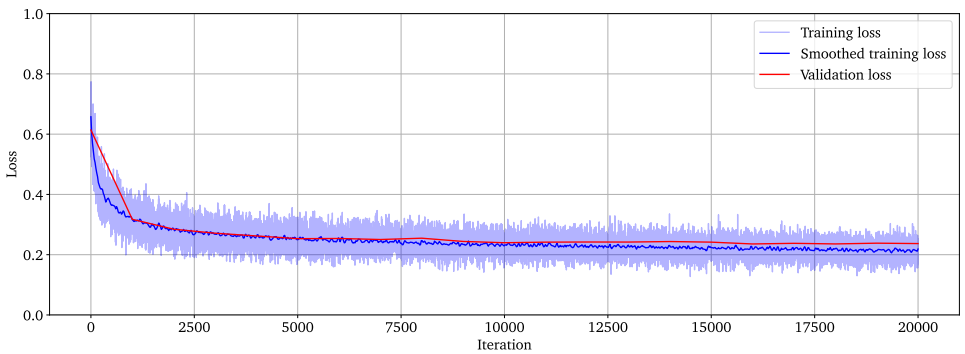
(b) *Angry*



(c) *Disgust*



(d) *Sleepy*



text, significantly impairing its ability to accurately capture and reproduce the intended content in those categories. The *Angry* ETTS model, however, was not affected because the angry speech in the dataset was clearer and did not include such non-verbal utterances.

Following the initial investigation, a similar study was conducted by other researchers, which claims the earlier findings. In this research, three manual preprocessing methods—resampling, silence trimming and removal of non-verbal utterances—were applied before the experiment [98]. After removing all non-verbal utterances, a remarkable improvement in both the intelligibility and emotional expressivity of the synthetic speech was observed, as measured by Word Error Rate (WER) and MOS evaluations. This confirms that acquiring and utilising a clear emotional speech dataset can significantly enhance the performance of an ETTS system when transfer learning is employed.

3.2.5 Dataset: ESD

The ESD dataset [99] emerges as a consequential stride towards establishing a robust, comprehensive and large emotional speech dataset aimed at enriching the linguistic and speaker variability among emotional speech datasets. The purpose of the ESD dataset was driven by a desire to improve the limitations of existing emotional speech datasets, and to establish a corpus that would enhance the performance and the scope of investigations in multi-speaker and multi-lingual ESS and other speech synthesis research.

The ESD dataset was designed for a multi-lingual nature, embodying both Chinese and English speech samples. This unique aspect provides strong support for investigating emotional expressions across diverse linguistic and cultural contexts. ESD demonstrates a remarkable breadth in speaker variability by incorporating emotional speech data from 10 native Chinese speakers and 10 native English speakers, where each includes 350 utterances across 5 emotion categories: *Neutral*, *Angry*, *Happy*, *Sad* and *Surprise* [99]. Such a configuration outperforms the speaker and lexical diversity provided in the previous datasets, thereby addressing the substantial need for them. The gender-balanced setting among the speakers further enhances the richness of data and facilitates in-depth exploration into speaker-independent ESS studies.

More importantly, to minimise confounding factors that could potentially affect performance, the ESD dataset maintains a uniform age range (25-35) among its speakers and requires the use of standard dialects—Standard Mandarin for Chinese speakers and North American English for English speakers [99]. Additionally, speakers were instructed to avoid any non-verbal utterances, such as laughter or sighs, during recording. This strict control over potential confounding factors enhances the fidelity and robustness of the ESD dataset.

Table 3.6: Information of the English Data in ESD

	Emotion Categories					Σ
	neu.	***amu.***	***ang.***	***dis.***	***sle.***	
Speakers (M/F)	5/5	5/5	5/5	5/5	5/5	5/5
Utterances (#)	350	350	350	350	350	350
Words (#)	2,203	2,203	2,203	2,203	2,203	11,015
Unique Words (#)	997	997	997	997	997	997
Total Duration (s)	9,135	9,800	10,430	9,450	9,555	48,370

neu.: *Neutral* ***amu.***: *Amused* ***ang.***: *Angry* ***dis.***: *Disgust* ***sle.***: *Sleepy*

The recording environment was carefully planned to achieve an optimal Signal-to-Noise Ratio (SNR) of over 20 dB with a sampling frequency of 16 kHz, ensuring that the recordings are suitable for state-of-the-art ESS frameworks [99]. Moreover, the organisation and partitioning of the ESD dataset are structured into training (300 utterances), evaluation (20 utterances), and test (30 utterances) sets, with non-overlapping utterances, making it a well-organised and ready-to-use resource for researchers.

As detailed in Table 3.6, the ESD dataset includes a total of 11,015 words, incorporating 997 unique lexical items across 1,750 utterances. This broad lexical coverage, which closely mirrors that of everyday communication, enhances the dataset's utility for ESS research [99]. Additionally, analyses of utterance duration and F_0 across different emotional categories provide valuable insights into the prosodic modulation of emotions, potentially serving as a catalyst for multilingual studies on ESS.

Collectively, the ESD dataset—with its multilingual scope, diverse speaker representation, controlled recording environment and robust organisational structure—serves as a valuable asset for advancing research in ESS. Through careful design and comprehensive statistical analysis, the ESD dataset not only addresses the gaps present in existing emotional speech datasets but also facilitates robust and nuanced investigations in the realm of emotional speech processing and analysis.

3.2.6 Transformer with ESD

The effectiveness of the ESD dataset was explored in the context of ETTS tasks employing transfer learning. To enhance the performance, a modification was applied to the network architecture by replacing the Tacotron 2 framework with a Transformer model. Although the Transformer-based model reached a similar level of performance in the MOS test as Tacotron 2, the Comparative MOS test revealed

an advantage for the Transformer-based framework, surpassing Tacotron 2 by a margin of 0.048 [40]. This section presents the setup and results of implementing the Transformer model with the ESD, while a detailed introduction to the Transformer model is provided in Section 5.2.1.

The architectural foundation of the model is based on the Transformer TTS framework [40], with the LJSpeech dataset [72] being used for the neutral TTS pre-training phase. The LJSpeech dataset contains 13,100 utterances, all spoken by a single female speaker, with each utterance carefully paired with its corresponding text transcription, forming a pair of *<wave, text>*. From a lexical perspective, the dataset includes a total of 225,715 words, with 13,821 unique lexical entities, averaging 17.23 words per utterance. The total duration of the dataset is approximately 23 h 55 m 17 s, with individual utterances ranging from 1.11 seconds to 10.10 seconds, averaging 6.57 seconds per utterance. The performance of LJSpeech has been validated through several TTS studies [98, 127, 128], affirming its suitability for this research.

The Transformer TTS architecture incorporates an additional linear layer to predict the stop token, which signals the model to stop synthesis when the stop token prediction reaches a value of 1 [40]. The length of the stop token vector matches the length of the synthetic Mel-spectrogram, where most values in the vector are 0, with the final digit being 1. A stop token weight, ranging from 5.0 to 8.0, is applied to the last digit during the computation of the loss in the training phase [40]. However, the repository [1] used in the experimental framework reported a problem with the stop learning upon the application of the stop token predictor, leading to the implementation of a hyperparameter related to maximum length to control the duration of the synthetic speech. To achieve automatic synthesis termination, a modification was made to the original loss function:

$$\mathcal{L}_{T-TTS} = \mathcal{L}_{mel} + \mathcal{L}_{post} + \lambda_{stop}\mathcal{L}_{stop} \tag{3.5}$$

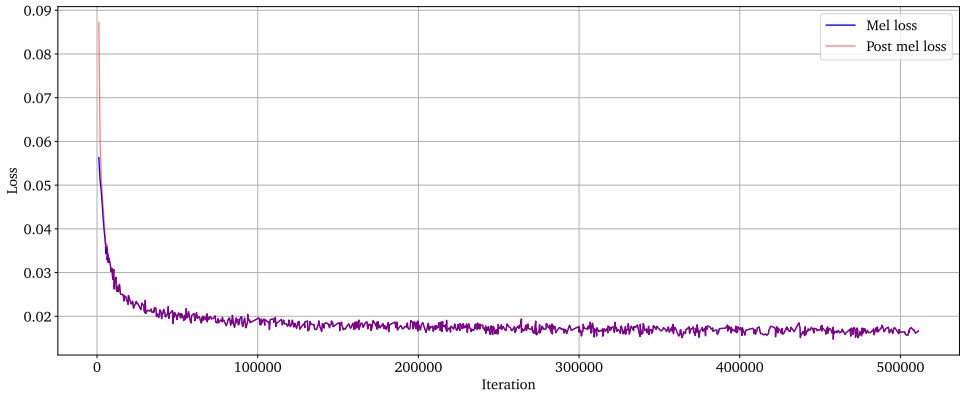
where the variables $\mathcal{L}_{mel}$ and $\mathcal{L}_{post}$ represent the MSE loss values computed using the outputs from the Transformer's decoder and the postnet along with the ground truth of the Mel-spectrogram, respectively. A weight λ_{stop} is applied to $\mathcal{L}_{stop}$, representing the binary cross-entropy loss associated with stop token prediction, with the optimal value determined to be 0.2 according to experimental evaluations.

The results of the neutral TTS system training are shown in Figure 3.5, in the form of loss curves. Ultimately, based on both achieved loss values and human perception, the model demonstrated excellent performance in both speech and duration prediction. After 450,000 iterations, the model achieved a $\mathcal{L}_{mel}$ of 0.015, a $\mathcal{L}_{post}$ of 0.015 and a $\mathcal{L}_{stop}$ of 2.034×10^{-5}. This model has been selected and is ready for fine-tuning through ETTS transfer learning.

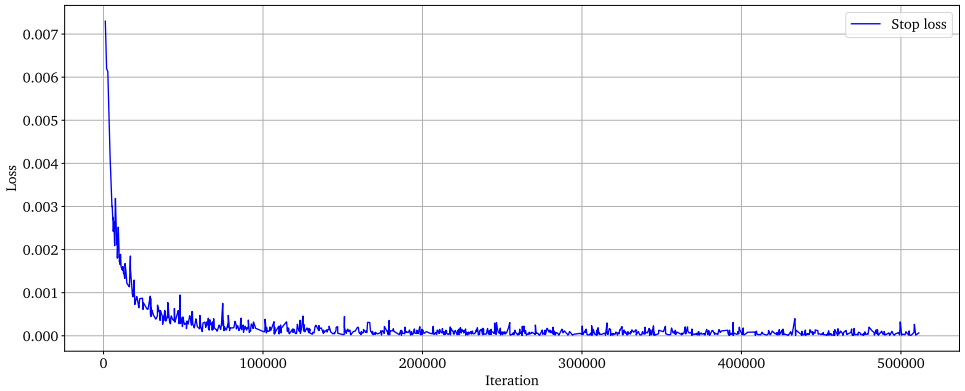
[1] `https://github.com/soobinseo/Transformer-TTS`

Figure 3.5: Loss Curves of Transformer-based TTS System

(a) $loss_{mel}$ and $loss_{post}$



(b) $loss_{stop}$



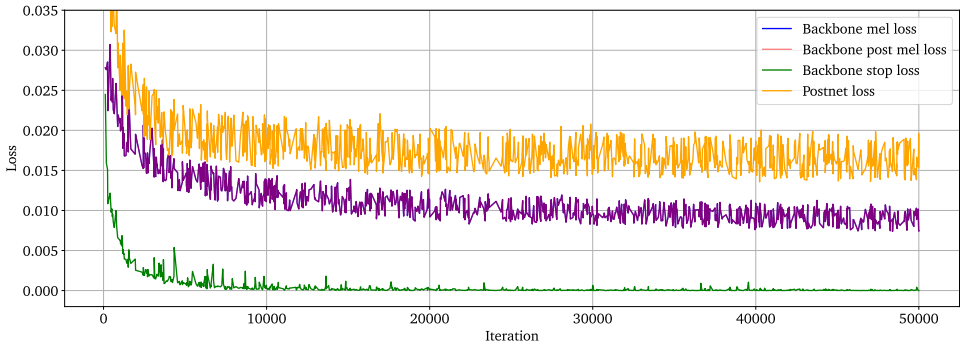
It is important to note that only the backbone of the Transformer was pre-trained using the LJSpeech dataset, as improvements were still needed in the prediction of the stop token. While the postnet in the neutral TTS system mentioned above was pre-trained by the author of the code, in the context of emotional transfer learning, both the backbone and the postnet require fine-tuning. Due to the limited number of samples in the ESD dataset and the emphasis on human perception, no validation set was used. Instead of computer-based validation, several samples were generated after every $2,000$ iterations for human evaluation.

Figure 3.6 illustrates the training results for both the backbone and postnet of the Transformer in synthesising speech across four emotional states: *Angry*, *Happy*, *Sad* and *Surprise*. The loss curves for both $\mathcal{L}_{mel}$ and $\mathcal{L}_{post}$ show significant oscillations, which can be attributed to the small size of the training set. All three losses stabled after $10,000$ iterations, and the best models for each emotional state were selected from the saved models after this point, based on human perception evaluations.

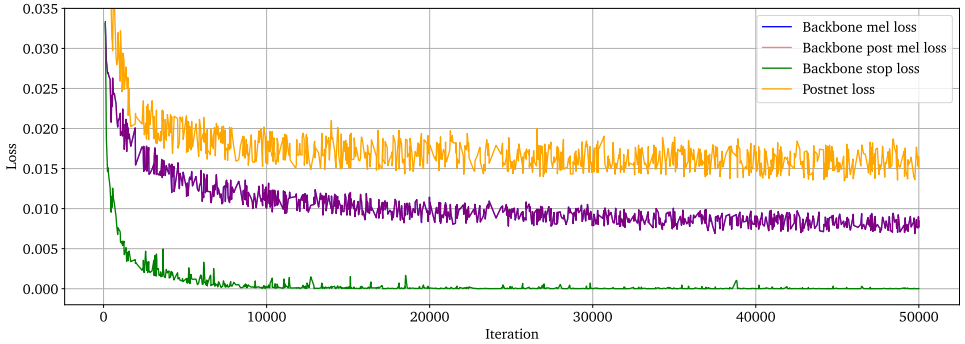
Unlike the ETTS system based on Tacotron 2 with the EmoV-DB dataset, described in Section 3.2.4, the generated speech across all four emotional categories demonstrates improved performance, not only in the *Angry* category. This outcome further confirms that a clearer emotional speech dataset, free from non-verbal utterances, can achieve better fidelity and intelligibility, leading to superior results in human perception evaluations.

Figure 3.6: Loss Curves of ETTS System of Transformer with ESD

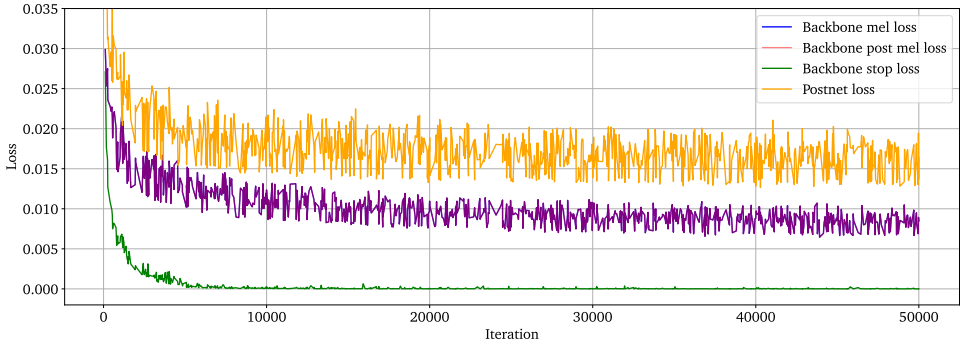
(a) *Angry*



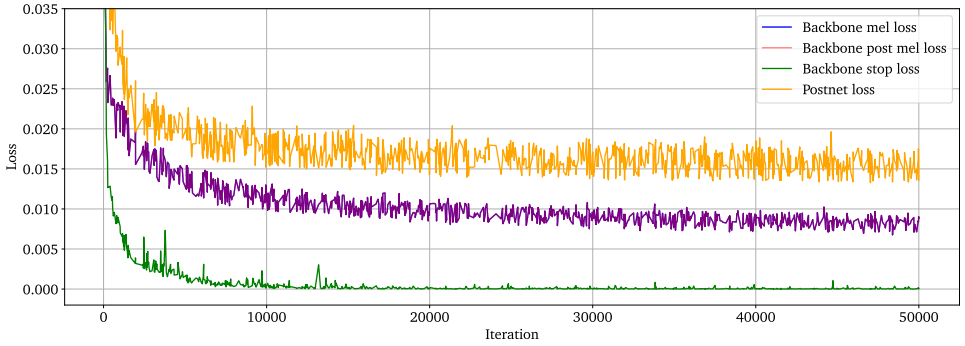
(b) *Happy*



(c) *Sad*



(d) *Surprise*



4

Frame-to-Frame Emotional Voice Conversion

In addition to the direct synthesis of emotional speech from text, an alternative methodology exists for generating emotional speech by utilising speech as the input, termed Emotional Voice Conversion (EVC). EVC constitutes a significant research field within ETTS, attracting significant attention from the research community. In this chapter, the background, definition, experimental investigations and results of EVC, will be discussed.

4.1 Voice Conversion and Emotional Voice Conversion

Voice Conversion (VC) represents an evolving technology with a pivotal function in the transformation of a speaker's vocal characteristics, whilst maintaining the underlying linguistic content [129]. More precisely, this technology facilitates the transformation of the vocal characteristics of one speaker into those of another speaker, thereby affording a diverse group of applicable and creative prospects. VC has shown its utility across a spectrum of domains, including speech synthesis, the realm of personalised virtual assistants, and the art of voice disguise [130].

4.1.1 Voice Conversion

In recent years, VC has undergone significant advancements, transitioning from conventional statistical methodologies to state-of-the-art deep learning techniques. Initially, VC relied mainly on statistical techniques such as Gaussian Mixture Models (GMMs) [131], Dynamic Kernel Partial Least Squares [132], Frequency Warping [133] and Non-Negative Matrix Factorisation [134]. These early methods were constrained by the need for parallel data, which required the availability of

corresponding utterances from both the source and target speakers. Despite the exploration of non-parallel data in the research of statistical techniques [135, 136], the rise of deep learning has declared significant improvements in the VC domain, making it more data-efficient and flexible.

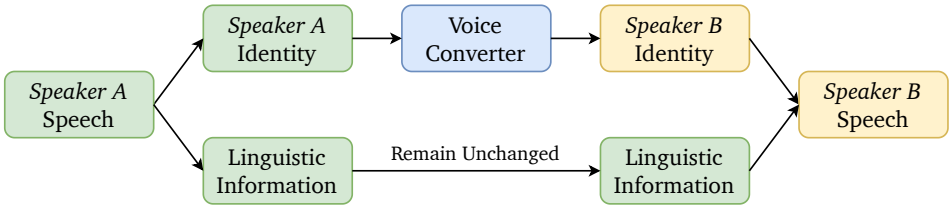
Deep learning, despite its inherent data-driven nature and its apparent dependence on the size of the dataset, has emerged as the key of contemporary VC research, primarily due to its efficacy in auxiliary tasks, such as speech reconstruction from converted features [130]. Notably, deep learning methodologies commonly applied in VC can be described based on their dependence on parallel training data. When leveraging parallel data, DNNs utilise their non-linear mapping capability to achieve the transformation of spectral features in frame-to-frame VC [137]. Additionally, temporal correlation among speech frames can be effectively exploited through the utilisation of LSTM-RNNs [138]. However, because of the constraint of the same duration between the source and converted speech frames in frame-to-frame VC, sequence-to-sequence models incorporating the 'encoder-attention-decoder' architectures have also been investigated to enhance performance in diverse variations of style among different speakers [139].

In order to the challenge posed by the lack of available parallel multi-speaker datasets, researchers have introduced four distinct categories of methods that operate independently of parallel data. The first approach involves the preservation of both linguistic and prosodic features during the process of speaker transformation [130], a task considerably more complex and difficult in the absence of parallel data. CycleGAN-VC [140] has raised as a notable success story in this regard. This method leverages a classical architecture of GAN, incorporating three different loss functions: adversarial loss, cycle consistency loss, and identity loss.

VC can further benefit from the potential of closely related technologies, notably TTS systems and ASR systems. The TTS module, characterised by its comprehension of linguistic content, offers valuable support for VC tasks, enabling the better preservation of linguistic information within the source speech [141]. Conversely, the ASR module supports VC tasks from a distinct perspective, emphasising the extraction of linguistic information from the source speech [142]. The capability of these modules is brought by their training on extensive corpora, thus the pre-trained modules free VC from the constraints of available dataset limitations.

Furthermore, another key focus within the realm of VC research involves the disentanglement of the speaker's identity from the linguistic content, based on the underlying assumption that speech is composed of both the speaker's vocal identity and the linguistic content [130]. This paradigm, by permitting the independent manipulation of the speaker's identity, introduces novel prospects for voice conversion. The architectural framework of the VC system based on the disentanglement of the speaker's identity is illustrated in Figure 4.1. Novel

Figure 4.1: Framework of VC System with Speaker's Identity Disentanglement



techniques, including Autoencoders [143] and GANs [144], have been effectively deployed to accomplish this disentanglement.

In the context of training data, it is worth noting that diverse resources have become valuable assets for VC research. Prominent among these are datasets such as CMU-Arctic [68] and VCTK [69], which include speech data from multiple speakers, making them feasible for both parallel and non-parallel VC tasks. Additionally, large-sized datasets, characterised by their extensive coverage, though potentially of lower quality (such as LibriTTS [74] and VoxCeleb [145]), play an important role in facilitating the training of fundamental components required for the investigation of VC.

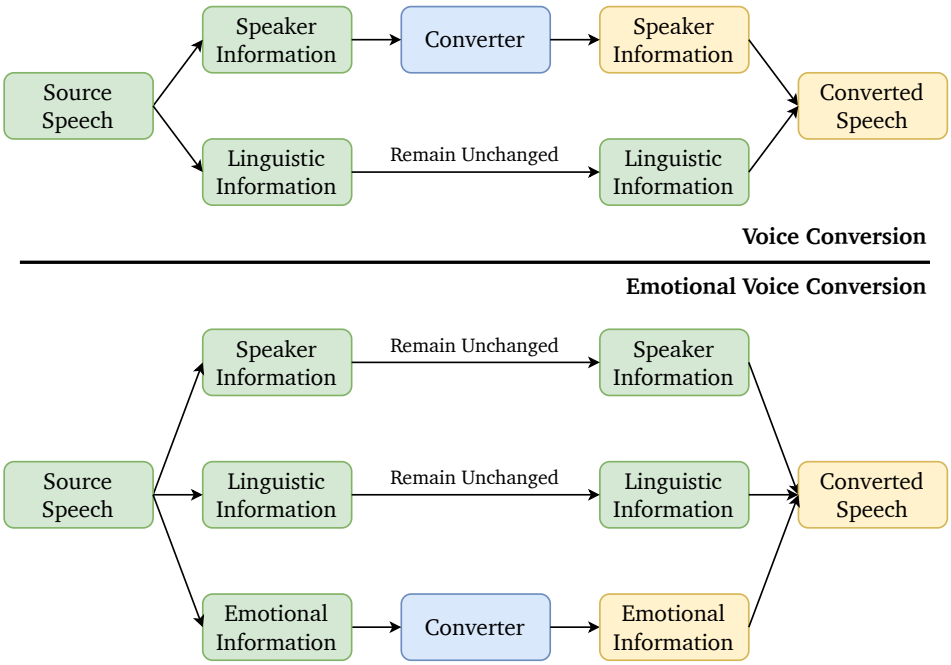
To summarise, VC stands as a dynamically evolving field with obvious practical applications. The technologies demonstrated above generally have the capacity to generate speech of superior quality, thereby substantially enhancing the efficiency of VC systems. The collaboration between deep learning methodologies and available multi-speaker datasets has opened up new opportunities, promising significant improvement in human-computer interactions and enabling creative voice-related applications.

4.1.2 Emotional Voice Conversion

As introduced in Section 4.1.1, VC, the well-established domain within the scope of speech processing, primarily centres its attention on the transformation of the speaker's identity while preserving the integrity of linguistic content [129]. Within the study of EVC, also known as Emotional Speech Conversion [146], researchers investigate under the premise of a two-piece assumption in VC, wherein speech contains a third element, namely emotional information. Consequently, EVC, being a specialised sub-field of VC, is dedicated to the transformation of emotional expression in speech, whilst preserving the integrity of both linguistic content and the speaker's identity [147]. The similarities and differences between VC and EVC are visually presented in Figure 4.2 for clarity and comprehension.

EVC represents an evolving and essential area within the realm of speech processing technology, playing a critical role in the enhancement of emotional expressivity in synthetic speech. Diverging from conventional TTS systems

Figure 4.2: Framework of VC System and EVC System



that primarily centre their efforts on the linguistic content of speech synthesis, EVC advances the field by incorporating emotional components into spoken communication [8]. This technological realm has acquired significant academic attention in recent years, as evidenced by the emerging research literature devoted to EVC. EVC holds essential relevance across various applications, including ETTS, SER and the development of conversational agents capable of expressing a wide range of emotional states [99].

A fundamental challenge encountered in the research of EVC is the accurate representation and manipulation of emotional prosody, a critical aspect of emotional expression in speech. Emotions in speech are expressed through various acoustic features, including pitch, energy and duration [99]. These acoustic features collectively produce emotional prosody, which plays an essential role in shaping listeners' perceptions and interpretations of the emotional information incorporated in the speech. Given the complexity of emotional prosody, characterised by the interplay of these acoustic features, the accurate characterisation and modelling of speech emotions bring a strong challenge. In the research of EVC, two vital research inquiries commonly arise: firstly, the effective description and representation of emotional states, and secondly, the modelling of the complex process including emotional expression and human perception [99].

Within the study of emotional representation, researchers have systematically investigated two principal approaches, similar to those employed in ETTS. The

categorical approach, inspired by foundational theories such as Ekman's six basic emotional states [148], namely *Anger*, *Disgust*, *Fear*, *Happy*, *Sad* and *Surprise*, discretely categorises emotions into distinct, predefined groups. While this categorical framework provides conceptual clarity, it is considered to lack the capacity to effectively capture the nuances of human emotions [99].

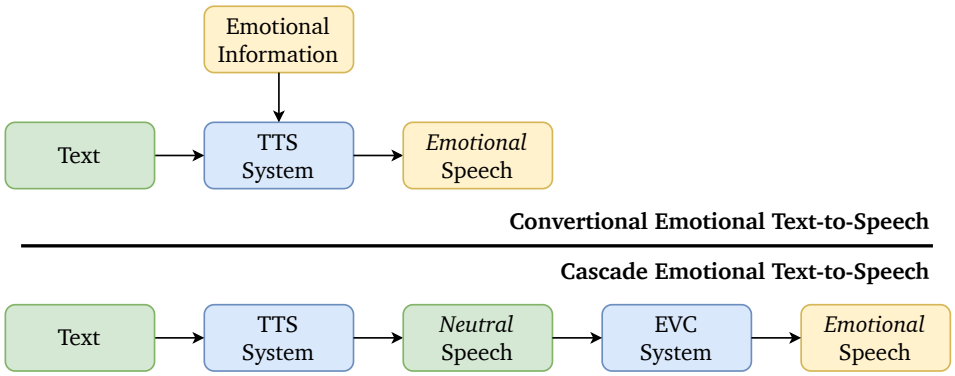
In contrast, the dimensional approach, exemplified by Russell's circumplex model [112], grounds emotions in dimensions like *arousal*, *valence* and *dominance*. This alternative approach provides a more nuanced and fine-grained representation of emotional states, enabling the distinction of subtle differences between closely related emotional states. Such dimensional representations play a pivotal role not only in the context of SER but also in the field of EVC, guiding the transformation of speech into different emotional states [99].

An additional important research challenge within EVC centres on the comprehensive modelling of the complex process of emotional expression and human perception, which is not involved in the research of VC. One proposal, which is frequently employed in the context of SER [149], is that the perception of emotion is essentially multi-layered, including various layers such as emotion categories, semantic primitives and acoustic features [99]. EVC, conversely, attempts to model the inverse process, wherein acoustic features are transformed to express multiple different emotional states. Notably, prosodic features, including voice quality, speech rate and F_0, hold particular significance in the research of EVC. F_0, in particular, assumes the importance of the characterisation of prosody, describing aspects of intonation and emotional expressiveness in speech [99]. Furthermore, on account of the inherently supra-segmental and hierarchical nature of emotional expression [150], emotions express themselves in both prosodic and spectral dimensions, a mapping that has typically not been a considered point in conventional VC research efforts [151]. Diverse methodological approaches, ranging from stylisation techniques [152, 153] to advanced deep learning approaches [154, 155], have been applied to manipulate F_0 and other acoustic features with the aim of achieving the target emotional state in speech.

The methodologies employed in the research of EVC can be further categorised based on their reliance on specific types of training data, presenting a distinction between Parallel and Non-Parallel approaches. Parallel techniques depend on the utilisation of paired or grouped utterances that maintain identical linguistic content and the speaker while differing in emotional expression [8]. This approach contributes to the ability of the model to understand the complex mapping from one emotional state to another. However, it is critical to acknowledge that the availability of parallel data resources is often limited and brings challenges in terms of quantity acquisition, especially when considering the requirements of deep learning methodologies.

Consequently, non-parallel methods have become advantageous alternatives in the research of EVC. Non-parallel data, wherein emotional expressions do not strictly

Figure 4.3: Architecture of Conventional and Cascade ETTS System



share the same linguistic content, presents a more scalable and practical solution [99]. Advanced techniques, such as Autoencoders [146] and CycleGANs [156], have effectively demonstrated the effectiveness of non-parallel approaches. These methods enable the acquisition of knowledge relevant to emotional domain transformation without the necessity of the expensive and restrictive parallel dataset.

In summary, EVC constitutes a promising and evolving research realm, offering the important potential for enhancing the expressiveness and emotional impact of synthetic speech. It effectively navigates the complexity of speech emotion, addressing the challenge of precise emotional prosody representation and manipulation. As technology continues to advance and novel methodologies arise, EVC stands on the edge of redefining our comprehension of and engagement with computer-generated speech, thereby leading to a paradigm shift in HCI by improving it with enhanced emotional expressiveness and context awareness.

4.1.3 Cascade ETTS with EVC

An additional aspect deserving attention is the relation between ETTS and EVC. The principal objective of ETTS is the synthesis of speech along with emotional expression based on provided textual sentences. Due to the fact that EVC has the capacity to convert the emotional state of the input speech to another emotional state whilst retaining other information such as linguistic features and speaker identity [78, 99], it is possible and practical to be investigated and implemented in the ETTS research and application. Despite its reliance on speech waveforms as input instead of textual sentences, EVC can be integrated as an auxiliary module within a neutral TTS framework, thereby realising a cascade TTS architecture as presented in Figure 4.3.

In a conventional ETTS system introduced in Chapter 2, the input text is processed by the model to generate the feature representation of emotional speech, which can subsequently be exploited by a vocoder, or alternatively, it

can directly generate the emotional speech as in an end-to-end ETTS system. However, in the cascade ETTS system presented in Figure 4.3, the neutral speech is initially synthesised by a TTS system, which is afterwards converted to emotional speech through the implementation of an EVC model. Despite the more complex architecture of the cascade ETTS system compared to its conventional application, the investigation of EVC has a significant advantage in leveraging the low-resource availability of emotional speech datasets and a discussion on this will be given in succeeding sections of this chapter.

4.1.4 Parallel and Non-Parallel EVC Approaches

Given the conceptual similarities between VC and EVC, the former being a more extensively explored and prominent subject within the research community, the investigation of EVC has naturally drawn inspiration and insights from VC. The methodologies employed in EVC can also be taxonomically categorised according to two primary criteria. First, classification by the model architecture includes two different categories, namely, Frame-to-Frame and Sequence-to-Sequence. Conversely, classification based on the requirement of training data includes parallel methods and non-parallel methods. In this chapter, we will commence by providing an overview of the more practical approaches performed by non-parallel training data first, subsequently followed by the parallel-data-based approaches, including the introduction, background, experiments, results and discussion.

As described in Section 4.1.2, the fundamental distinction between parallel and non-parallel datasets predominantly depends on the requirement for identical linguistic content across different emotional categories. The inherent characteristics of parallel datasets pose substantial challenges and constraints during their collection. The recording process requires that a single actor performs the same sentence in various emotional states, thereby resulting in considerable workloads for the performers, alongside increasing time and financial burden for the data collectors. In fact, upon a comprehensive survey of the field of emotional speech datasets, it is evident that the popularity of non-parallel datasets significantly surpasses that of parallel datasets.

Another influencing factor that illustrates the observation above involves the field of affective computing. The majority of investigations within this field are primarily oriented towards affect recognition tasks, including the research of emotion recognition and sentiment analysis [9, 157]. Furthermore, it is noteworthy that within the TTS research, the integration of emotional expression into synthetic speech is not commonly addressed. Hence, the availability of speech datasets that satisfy the requirements for EVC studies based on parallel training data—even those relying on parallel training data—remains considerably constrained.

Another related concern regarding the training data employed in the EVC research deserves much attention. For the purpose of enhancing the performance

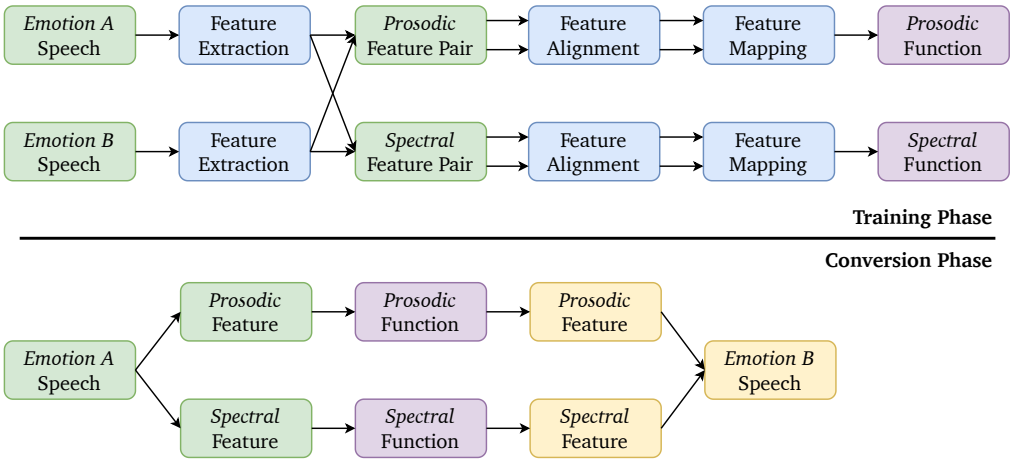
and robustness of SER systems, training datasets often incorporate irrelevant noise alongside the speech signal [103, 104]. Furthermore, certain datasets containing conversational speech samples, such as the IEMOCAP dataset [103], intentionally include samples of overlapping speech involving two speakers. These intentional contents are designed to simulate real-world scenarios, thereby assisting the model in achieving better performance across diverse and complex real-life environments. However, it must be noted that the model training for both EVC and ETTS do not derive advantages from this characteristic; rather, they are held back by it. The model training for EVC and ETTS requires original, clear and high-quality speech samples, carefully recorded using professional equipment in controlled studio environments [99]. Consequently, the situation relating to the lack of suitable training data for both EVC and ETTS, be it through parallel or non-parallel approaches, is aggravated by the reasons above.

4.2 Conventional Frame-to-Frame EVC

As discussed in Section 4.1.4, the utilisation of EVC methodologies that rely on non-parallel data is more practical and has gained attention within the research community, due to the compelling reasons described. An in-depth investigation of state-of-the-art highlights that frame-to-frame EVC methodologies exhibit remarkable potential in leveraging non-parallel training data, thereby demonstrating commendable performance. However, these compromise approaches are not free from their inherent limitations. The most important of these limitations is the poor compatibility of conventional frame-to-frame EVC techniques with non-parallel training data. The conventional frame-to-frame EVC approach typically includes three main stages: Feature Extraction, Feature Alignment and Feature Mapping [99]. Figure 4.4 illustrates the architectural framework characterising the conventional frame-to-frame EVC system.

In the conventional frame-to-frame EVC methodologies, two distinct speech samples called Source speech and Target speech, each characterised by different emotional expressions, are utilised as input data. Subsequently, Prosodic and Spectral features are extracted from both the source speech and target speech through the utilisation of an identical feature extraction technology. This common feature extraction procedure is then followed by the respective alignment of prosodic and spectral features for both source and target speech samples. Following this alignment step, the mapping between the source and target spectral features is modelled, and another model is implemented to learn the corresponding mapping between the prosodic features. Consequently, two distinct mapping functions are acquired, which facilitate the conversion of both prosodic and spectral features from the source emotional state to the target emotional state. In the conversion phase, the converted prosodic and spectral features serve as input to a vocoder, enabling

Figure 4.4: Framework of Conventional Frame-to-Frame EVC System



the generation of the converted speech. It must be emphasised that due to the nuanced nature of emotional expression, this conversion process is different from conventional VC, which exhibits supra-segmental and hierarchical characteristics. Consequently, both prosodic and spectral features must be converted to capture the full scope of emotional expression from the source emotional state to the target state [99]. Hence, the initial step in EVC requires the extraction of prosodic and spectral features from both the source and target speech samples.

4.2.1 Feature Extraction

In the domain of spectral features commonly employed within frame-to-frame EVC investigations, Mel-frequency-based cepstral features have emerged as a predominant choice, as evidenced by a large number of studies [146, 147, 151, 154, 155, 156, 158, 159, 160, 161, 162, 163, 164, 165]. It is notable that Mel-frequency-based cepstral features, commonly referred to as MFCCs and MCEPs, exhibit considerable similarities in terminology and have often been interchangeably referenced in academic literature [166, 167]. Both methodologies involve cepstral analysis and serve as representations of the spectrum, yielding highly relevant features. However, the primary difference between them is the extraction process. In practice, MFCCs are derived through the implementation of a Mel filter bank [168, 169], whereas MCEPs are obtained from Smoothed Spectral Envelopes (SSEs) to achieve similar results [170, 171]. Consequently, in this paper, the application of a Mel filter bank serves as a defining criterion for distinguishing between MFCCs and MCEPs.

Beyond their application within TTS research, as introduced in Section 2.1.4, MFCCs and MCEPs have demonstrated excellent performance across various domains in speech-related studies, notably in ASR [5] and SER [172]. The

computation of MFCCs involves a sequential four-step process: initialisation with the computation of the power spectrum from the input speech signal, application of a Mel filter bank—engineered to mimic the human auditory system's response [173]—onto the spectrum, logarithm computation of the powers at each Mel-frequency band, and subsequent decorrelation via Discrete Cosine Transformation (DCT) [174]. One example of Mel-filters with triangular overlapping windows is

$$Mel(f) = 2595log(1 + \frac{f}{700}) \tag{4.1}$$

where f represents the original frequency [174]. Toolkits like librosa [168], OpenSMILE [175] and WORLD [53], are widely employed for the extraction of MFCCs and MCEPs in state-of-the-art research. Additionally, the 80-dimensional Mel-spectrogram has also demonstrated effectiveness in representing spectral features within emotional speech for frame-to-frame EVC [176], as previously introduced in Section 2.1.4.

In the field of frame-to-frame EVC studies, the utilisation of F_0 stands out as a universal prosodic feature, occupying prominently in nearly all state-of-the-art investigations. Exploiting the information embedded within F_0, Continuous Wavelet Transformation (CWT) serves as a consistent approach, facilitating the decomposition of the F_0 contour into distinct temporal scales. This enables the modelling of diverse prosodic levels ranging from short-term to long-term variations [158]. It is worth noting that the efficacy of CWT has been confirmed across various fields of audio signal research [177, 178, 179, 180]. When applied to an input signal, such as the F_0, CWT can be mathematically expressed as:

$$W(s,p) = \frac{1}{\sqrt{s}} \int_{-\infty}^{+\infty} F_0(t) \cdot \psi(\frac{t-p}{s}) dt \tag{4.2}$$

Here, the symbol ψ denotes a continuous function referred to as the mother wavelet, instrumental in generating daughter wavelets through operations involving scaling by using the scale parameter s and translation by using the position parameter p. An example of a customary mother wavelet is the Mexican hat wavelet, determinable via calculations involving:

$$\psi_M(t) = \frac{2}{\sqrt{3}} \pi^{-\frac{1}{4}} \cdot (1 - t^2) \cdot exp(-\frac{t^2}{2}) \tag{4.3}$$

However, in the realm of signal processing, the common practice is the adoption of Discrete Wavelet Transformation (DWT) instead of CWT, employing numerous sampling scales for effective implementation [161]. The mathematical form representing DWT is expressed as:

$$\tilde{W}(j,k) = \int_{-\infty}^{+\infty} F_0(t) \cdot \psi_{j,k}(t) dt \tag{4.4}$$

$$\psi_{j,k} = \frac{1}{\sqrt{2^j}} \psi \cdot (\frac{t - 2^j k}{2^j}), j = 1, 2, 3, ..., J \tag{4.5}$$

where the symbol 2^j denotes the sampling scale, while J signifies the number of discrete scales employed. The parameter $k \in \mathbb{Z}$ represents the translation parameter, as the translation usually exhibits a proportional relationship to the scale within DWT. Additionally, it is noteworthy that the inverse transformation is able to achieve an approximate reconstruction of the input F_0 [161].

$$F_0(t) \approx \sum_{j=1}^{J} \tilde{W}(j,k) \cdot \psi_{j,k}(t) \tag{4.6}$$

Typically, the determination of the number of discrete scales is contingent upon the octave configuration within the system. For instance, a 10-scale setup implies that each scale corresponds to one complete octave [156], while a 30-scale configuration signifies that each scale corresponds to one-third of an octave [165]. Moreover, investigations within the field of EVC have explored adaptive scaling methodologies within CWT specifically applied to F_0 [160].

Indeed, CWT represents only one approach among various methodologies employed to extract valuable information from F_0. DCT [181] and statistical features encompassing measures such as mean, topline and baseline [182] of F_0 have demonstrated exceptional performance within the EVC research. Furthermore, among a series of prosodic features incorporated in frame-to-frame EVC research, the spectrum, denoted as $\hat{f}(\xi)$ obtained from the input speech $f(t)$ through Equation 2.7, the energy of which has also been considered [147, 159]. The computation of the energy of the spectrum in frames can be expressed as:

$$E_t = \sqrt{\sum_{d=1}^{D} \left| \hat{f}(\xi)_{d,t} \right|^2}, t = 1, 2, 3, ..., T \tag{4.7}$$

The spectrum, denoted as $\hat{f}(\xi)$, is characterised by dimensions represented as (D, T), where D expresses the feature dimension, and T corresponds to the number of frames within the spectrum [159]. Additionally, CWT presents an alternative methodology for leveraging the information of the energy contour across various temporal scales.

4.2.2 Feature Alignment

In the pursuit of aligning identical feature sets within both the source and target speech signals, the commonly applied method employed in frame-to-frame EVC

investigations is Dynamic Time Warping (DTW) [158, 160, 163, 165, 183]. DTW represents a widely recognised technique used for measuring the similarity between two sequences and subsequently aligning them by temporal transformations to fit the detected similarities in their shapes [184]. The application of DTW needs to comply with three constraints for the given sequences:

1. Each element within one sequence must be matched with at least one corresponding element in the other sequence.
2. The initial and final elements of each sequence should be matched with the initial and final elements of the other sequence, respectively.
3. The indices representing matched elements in one sequence and their corresponding indices in the other sequence must be monotonic increasing.

All speech pairs within the parallel emotional speech dataset inherently satisfy the constraints above. However, the applicability of DTW is limited in research employing non-parallel training data due to the reliance on sequence similarity comparisons, making it unsuitable for such datasets due to different linguistic content within paired samples. Algorithm 1 describes the procedural steps of classical DTW, which primarily involves the computation of the accumulated cost matrix.

where the symbol $dtw_{i,j}$ denotes the distance measure between $\mathbf{S}_{1:i}$ and $\mathbf{T}_{1:j}$, signifying the cumulative distance between the sequences from the beginning to the indices i and j. Hence, $dtw_{m,n}$ represents the cumulative distance between the source sequence $\mathbf{S}$ and the target sequence $\mathbf{T}$. Besides, recent advancements in neural networks have revealed several promising methods for feature alignment. These methods include novel techniques such as speech recognisers [185] and attention mechanisms [186, 187].

4.2.3 Feature Mapping

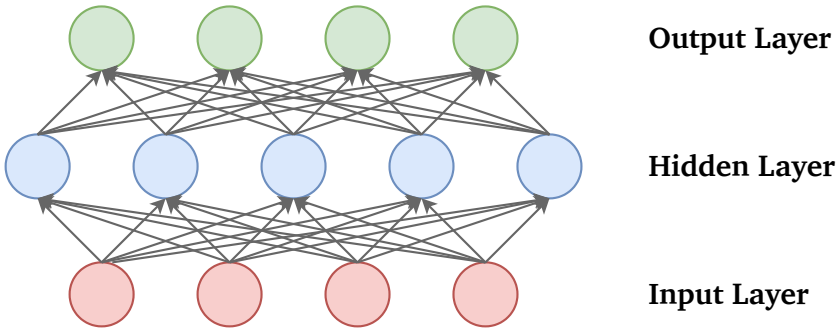
As shown in Figure 4.4, subsequent to the feature alignment process, it follows the important phase of discovering the relationship between the source and target features, commonly referred to as feature mapping. Early exploration within conventional frame-to-frame EVC approaches involved employing statistical modelling to attain feature mapping. One of the classical statistical model, GMM, has been utilised in various studies. Notably, GMMs were employed to convert statistical F_0 features on a syllable-wise basis [182], while another study utilised two GMMs to individually convert the spectrum and DCT coefficients of F_0 [181]. Furthermore, the adoption of Non-negative Matrix Factorisation contributed to the conversion of both spectrum and F_0, presenting enhanced performance in emotional expressivity through a MOS test compared to individual conversions, as described in prior studies [158, 183].

Algorithm 1: Computation of Accumulated Cost Matrix by DTW

Input: Source Sequence: $\mathbf{S} = (s_1, s_2, s_3, \cdots, s_m)$, $m \in \mathbb{R}^+$
Target Sequence: $\mathbf{T} = (t_1, t_2, t_3, \cdots, t_n)$, $n \in \mathbb{R}^+$
Output: Accumulated Cost Matrix: dtw

$dtw \leftarrow \mathcal{M}^{(m+1)\times(n+1)}$
$i \leftarrow 0$
while $i \leq m$ **do**
  $j \leftarrow 0$
  while $j \leq n$ **do**
    $dtw_{i,j} \leftarrow \infty$
    $j \mathrel{+}= 1$
  end
  $i \mathrel{+}= 1$
end
$dtw_{0,0} \leftarrow 0$
$i \leftarrow 1$
while $i \leq m$ **do**
  $j \leftarrow 1$
  while $j \leq n$ **do**
    $dtw_{i,j} \leftarrow \|\mathbf{S}_i - \mathbf{T}_j\| + min(dtw_{i-1,j}, dtw_{i,j-1}, dtw_{i-1,j-1})$
    $j \mathrel{+}= 1$
  end
  $i \mathrel{+}= 1$
end
return dtw

Figure 4.5: Architectural Framework of MLP



4.2.3.1 Multi-Layer Perceptron

In recent years, the advancement in neural network technology has presented convincing prospects to achieve feature mapping. This exploration spans from the employment of fundamental neural networks such as DBNs and Multi-Layer Perceptrons (MLPs) [154, 163, 165]. Within this context, DBNs were utilised for spectral feature conversion (specifically, MCEPs), while MLPs were deployed for prosodic feature conversion (specifically, CWT-F_0). An MLP architecture is composed of multiple fully-connected layers, thereby often referred to as a Fully-connected Network. The architectural representation of an MLP is given in Figure 4.5. The computational process within each neuron of an MLP with inputs can be expressed as:

$$y = a(\sum_{i=1}^{I} w_i \cdot x_i + b) \tag{4.8}$$

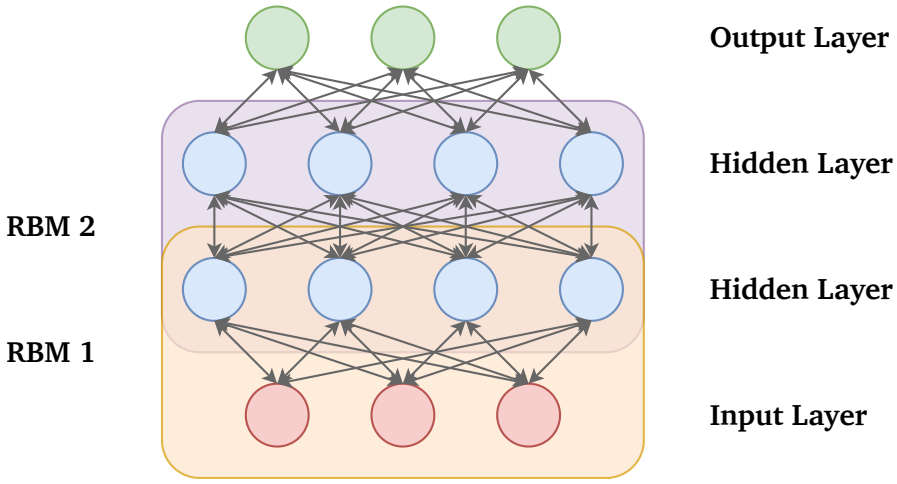
where the symbols x_i and w_i denote the value and weight, respectively, of the i-th input neuron. Here, I indicates the total number of input neurons, while b represents the bias of the current neuron. The symbol a refers to the activation function employed to introduce non-linear characteristics to the linear computation and regulate the output value range. As an example, a classical activation function is the hyperbolic tangent function, denoted as Tanh, and computed by the formula:

$$tanh(x) = \frac{exp(x) - exp(-x)}{exp(x) + exp(-x)} \tag{4.9}$$

4.2.3.2 Deep Belief Network

A DBN [154, 163, 165] is constructed from multiple Restricted Boltzmann Machines (RBMs), each comprising one visible layer and one hidden layer, visually represented in Figure 4.6. Unlike MLPs, RBMs facilitate bidirectional propagation, allowing

Figure 4.6: Architectural Framework of DBN



interactions between the visible and hidden layers. Consequently, each neuron within the visible layer possesses an activation probability expressed as:

$$P(v_i = 1|h) = \sigma(\sum_{j=1}^{J} w_{ij} \cdot h_j + b_i) \tag{4.10}$$

where $P(v_i = 1|h)$ denotes the probability of activation for the i-th visible neuron, b_i denotes the bias of the i-th visible neuron, and w_{ij} represents the weight between the i-th visible neuron and the j-th hidden neuron. Besides, h_j stands for the value of the j-th hidden neuron, and J indicates the total number of hidden neurons. The symbol σ corresponds to the *sigmoid* activation function, mathematically expressed as:

$$\sigma(x) = \frac{1}{1 + exp(-x)} \tag{4.11}$$

Likewise, the probability of activation for the j-th hidden neuron can be computed as:

$$P(h_j = 1|v) = \sigma(b_j + \sum_{i=1}^{I} w_{ij} \cdot v_i) \tag{4.12}$$

In this equation, b_j represents the bias of the j-th hidden neuron, while v_i denotes the value of the i-th visible neuron, and I denotes the total number of visible neurons.

4.2.3.3 Recurrent Neural Network

Several other neural network architectures have been employed in the latest research. Despite the full connectivity of MLPs, their structure is not qualified to consider sequential or temporal information inherent in input speech signals. Therefore, RNNs have attracted significant attention and application across various tasks related to speech signals [188, 189]. Neurons within RNNs compute the hidden state h_t not solely based on their current input x_t, but also depend on the hidden state from the previous timestep h_{t-1}. Mathematically, this computation is represented as:

$$h_t = a(w_{hh} \cdot h_{t-1} + w_{xh} \cdot x_t + b_h) \tag{4.13}$$

where the symbol w_{hh} denotes the weight parameter between the two hidden states, h_{t-1} and h_t, while w_{xh} represents the weight parameter connecting the current input x_t to the hidden state h_t. Additionally, b_h denotes the bias, and a signifies the activation function used in the computation. To compute the output at the current timestep:

$$y_t = w_{hy} \cdot h_t + b_y \tag{4.14}$$

where w_{hy} represents the weight parameter connecting the hidden state to the output, while b_y denotes the bias associated with the output computation.

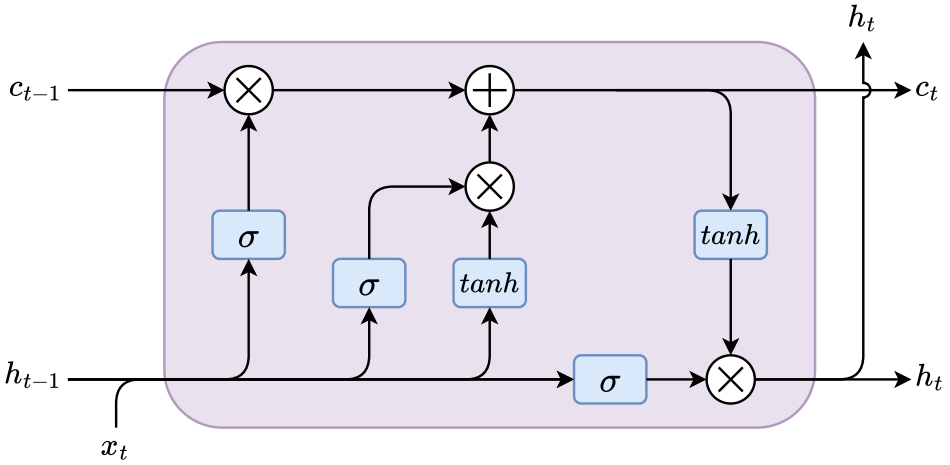
Despite the advantages, conventional RNNs face two significant challenges: the long-term dependency problem, where neurons from later timesteps struggle to memorise the information from earlier ones, and issues during training, wherein computed errors tend to either escalate or diminish across timesteps in backpropagation [190]. This latter phenomenon, known as Gradient Explosion or Gradient Vanishing, is particularly evident as sequences grow longer [191]. Mitigating gradient issues during backpropagation often involves employing L_1 and L_2 regularisation techniques. However, models employing LSTM address these challenges differently by adapting the structure of conventional RNN neurons to benefit the preservation of long-term dependencies and solve gradient explosion and vanishing problems simultaneously. The architecture of an LSTM cell is presented in Figure 4.7.

The LSTM mechanism assigns a cell state and three distinct gates to each cell. Initially, upon receiving the current input x_t and the preceding hidden state h_{t-1}, a forget gate f_t is employed to decide which information from the preceding cell should be forgotten:

$$f_t = \sigma(w_{fh} \cdot h_{t-1} + w_{fx} \cdot x_t + b_f) \tag{4.15}$$

In this equation, w_{fh}, w_{fx}, and b_f represent the weights associated with the previous hidden state, the current input, and the bias, respectively. Subsequently,

Figure 4.7: Architectural Framework of LSTM



similarly to the conventional RNNs, new information from the input x_t and the preceding hidden state h_{t-1} are employed to compute the input gate i_t. Thus, the input gate manages what the information should be received at the current time-step:

$$i_t = \sigma(w_{ih} \cdot h_{t-1} + w_{ix} \cdot x_t + b_i) \tag{4.16}$$

where symbols w_{ih}, w_{ix}, and b_i denote the weights associated with the previous hidden state, the current input, and the bias, respectively. The subsequent step involves computing the candidate value $\tilde{c}_t$ for the cell update using h_{t-1} and x_t:

$$\tilde{c}_t = tanh(w_{ch} \cdot h_{t-1} + w_{cx} \cdot x_t + b_c) \tag{4.17}$$

where symbols w_{ch}, w_{cx}, and b_c represent the weights associated with the previous hidden state, the current input, and the bias, respectively. Moreover, in conjunction with the input gate i_t, forget gate f_t, and the cell state from the previous cell c_{t-1}, the current cell state can be updated and obtained:

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \tag{4.18}$$

To generate the hidden state for the current timestep, an output gate o_t is initially computed using the previous hidden state h_{t-1} and the input x_t:

$$o_t = \sigma(w_{oh} \cdot h_{t-1} + w_{ox} \cdot x_t + b_o) \tag{4.19}$$

where symbols w_{oh} and w_{ox} denote the respective weights, while b_o represents the bias. Ultimately, the hidden state is updated as follows:

$$h_t = o_t \cdot tanh(c_t) \tag{4.20}$$

Hence, the output value of the current cell can be obtained using Equation 4.14.

However, another limitation of RNNs lies in their ability to leverage information. Conventional RNNs and LSTM-RNNs can only access and process information from preceding contexts, whereas the information from future contexts could also be leveraged to enhance speech processing since the entire speech signal is processed at the same time [192]. Consequently, a modified RNN architecture capable of processing sequences in both forward and backward directions was introduced, known as Bidirectional RNN [193]. Given Equation 4.14, the alteration to the output of the current timestep is expressed as:

$$y_t = w_{\overrightarrow{h}y}\overrightarrow{h}_t + w_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y \tag{4.21}$$

which describes the bidirectional nature of information flow, with two types of arrows representing the forward and backward directions. In frame-to-frame EVC investigations, bidirectional LSTM-RNNs demonstrated commendable performance converting MCEPs, CWT-F_0, and the energy of spectrum [159].

4.2.4 Non-Parallel Training

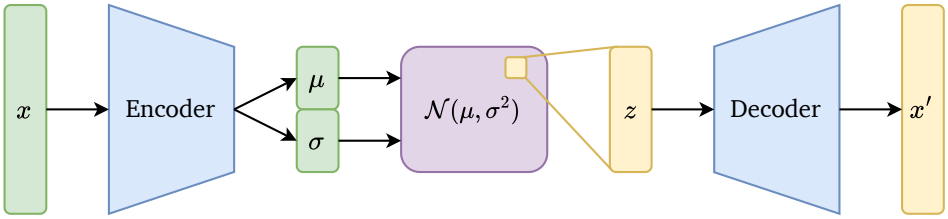
However, within the framework of conventional frame-to-frame methods, feature alignment and mapping face limitations when applied to non-parallel training data due to inconsistencies in the linguistic content. Therefore, alternative strategies have been proposed to address the challenges caused by the demanding and costly collection of parallel data. These include the application of emotional domain translation, disentangling linguistic and emotional information, and the integration of TTS and ASR systems as potential solutions [99]. Moreover, researchers have deeply researched two neural network architectures, namely Variational Autoencoder (VAE) and GAN, within the domain of non-parallel frame-to-frame EVC research.

4.2.4.1 Variational Autoencoder

In the researcher of EVC, VAEs have become a potent mechanism, introducing a novel perspective to the manipulation of emotional features within speech signals. Autoencoder, a type of unsupervised neural network consisting of an Encoder and a Decoder, aims to learn latent representations from the input [194]. The encoder compresses the input into a lower-dimensional space, while the decoder subsequently reconstructs the original input from the latent representations. This sequence of processes can be represented as:

$$\begin{aligned} z &= enc(x) \\ x' &= dec(z) \end{aligned} \tag{4.22}$$

Figure 4.8: Architectural Framework of VAE



where the measure $|x - x'|^2$ represents the distance between the initial input x and the reconstructed input x', thereby denoting the performance of the latent representation z.

Nevertheless, a significant challenge arises when employing Autoencoders for generative purposes. While Autoencoders have the capability of generating x' directly from the representation z, the breadth of diversity and novelty in the generated content is restricted by the deterministic encoding nature of Autoencoders. Hence, VAEs diverge from this deterministic approach by employing an encoder that produces the posterior probability of a Gaussian distribution $q_\phi(z|x)$ [195].

$$q_\phi(z|x) = \mathcal{N}(z|\mu_\phi(x), \sigma^2_\phi(x)) \tag{4.23}$$

where μ represents the mean and σ denotes the standard deviation. Additionally, the decoder is also probabilistic in nature:

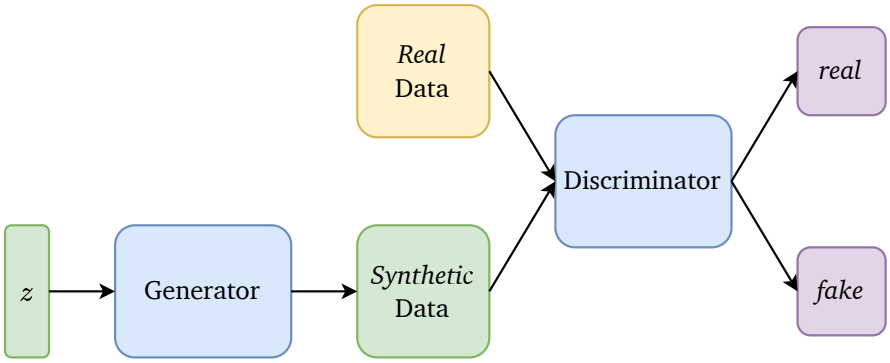
$$p_\theta(x|z) = \mathcal{N}(x|\mu_\theta(z), \sigma^2_\theta(z)) \tag{4.24}$$

The architecture of VAEs is described in Figure 4.8. This inherent probabilistic nature provides VAEs with the capability to generate content. Notably, in the research of EVC, an adapted VAE model that learns a factorised representation from features across multiple temporal scales within the input speech effectively transformed the Mel-spectrogram from one emotional state to another [176].

4.2.4.2 Generative Adversarial Network

The application of GANs brought a revolutionary paradigm within generative modelling. The GAN architecture consists of two neural networks: a Generator, denoted as G, and a Discriminator, denoted as D, engaged in an adversarial training process resembling a competitive game. As illustrated in Figure 4.9, the generator produces synthetic data by utilising a prior distribution $p_z(z)$, attempting to approach the distributions $p_{data}(x)$ representing real data x. Conversely, the role of the discriminator is to distinguish between the original and the synthetic samples, or rather, distinguish *real* and *fake* samples [196]. Essentially, the generator

Figure 4.9: Architectural Framework of GAN



aims to generate new samples conforming to the learned distribution, while the discriminator takes responsibility of identifying the authenticity of the generated samples, attempting for them to resemble real-world data. This interplay between the generator and the discriminator involves a two-player minimax game controlled by the value function $V(G, D)$ [197], formulated as:

$$\min_G \max_D V(G, D) = \mathbb{E}_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})}[log\, D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[log\,(1 - D(G(\boldsymbol{z})))] \quad (4.25)$$

This value function serves as a catalyst for the ongoing enhancement of both networks, benefiting the creation of remarkably realistic and diverse content. GANs have exhibited exceptional performance across multiple fields, including CV [198], CA [199] and NLP [200]. Their ability to generate novel content of superior quality has propelled them to the forefront of advanced research in AI and machine learning.

Numerous GAN variants have been proposed, and several of which have been implemented in frame-to-frame EVC studies, demonstrating outstanding efficacy. Combined with fully-convolutional networks, a dual adversarial network architecture effectively transformed MCEPs and CWT-F_0 from neutral speech into emotional speech, incorporating expressions like *Happiness*, *Anger* and *Sadness* [161]. Additionally, leveraging the encoder-decoder structure of Autoencoders, various Autoencoder variants have exhibited their effectiveness when integrated with GANs, such as Autoencoder GAN [155], Variational Autoencoder GAN (VAE-GAN) [146] and Variational Autoencoding Wasserstein GAN (VAW-GAN) [151, 164]. Furthermore, alternative architectures like CycleGAN [147, 156] and its advanced model, StarGAN [162], have achieved remarkable performance in frame-to-frame EVC tasks.

4.3 Application: CycleGAN

As discussed in Section 4.2 and 3.2, the practical application of EVC using the EmoV-DB dataset, which contains a limited number of parallel samples, implements non-parallel training. This method allows the utilisation of each speech sample for training purposes without requiring its corresponding match or the exclusion of samples lacking counterparts in other emotional states. Furthermore, recent state-of-the-art research validates the feasibility of speaker-independent training in non-parallel EVC investigations [151], where speech samples from multiple speakers are combined as the training set, significantly expanding the pool of available training sample pool. Consequently, this section introduces CycleGAN, one of the frame-to-frame EVC solution based on adversarial training.

4.3.1 CycleGAN

The conventional GAN architecture encompasses a generator and a discriminator, responsible for content generation and real-vs-fake discrimination, respectively. CycleGAN, initially developed for image style transfer in CV, addresses the challenge of insufficient parallel training samples [201]. Its efficacy in style transfer has also been validated in the research of VC [140, 202]. The highlight of CycleGAN, beyond the standard source-to-target mapping, is the introduction of an inverse mapping with cycle consistency loss, to optimise the mapping without the corresponding sample. As shown in Figure 4.10, this alteration on regular GAN architecture involves the deployment of two generators $G_{X\to Y}$ and $G_{Y\to X}$, along with two discriminators D_X and D_Y, a deviation from the conventional single generator and discriminator setup of GAN.

Figure 4.10a illustrates the regular adversarial training process, where a slight modification is made on Equation 4.25:

$$\begin{aligned}\mathcal{L}_{adv}(G_{X\to Y}, D_Y) = &\mathbb{E}_{\boldsymbol{y}\sim p_{data}(\boldsymbol{y})}[log\, D_Y(\boldsymbol{y})] \\ &+ \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[log\,(1 - D_Y(G_{X\to Y}(\boldsymbol{x})))]\end{aligned} \tag{4.26}$$

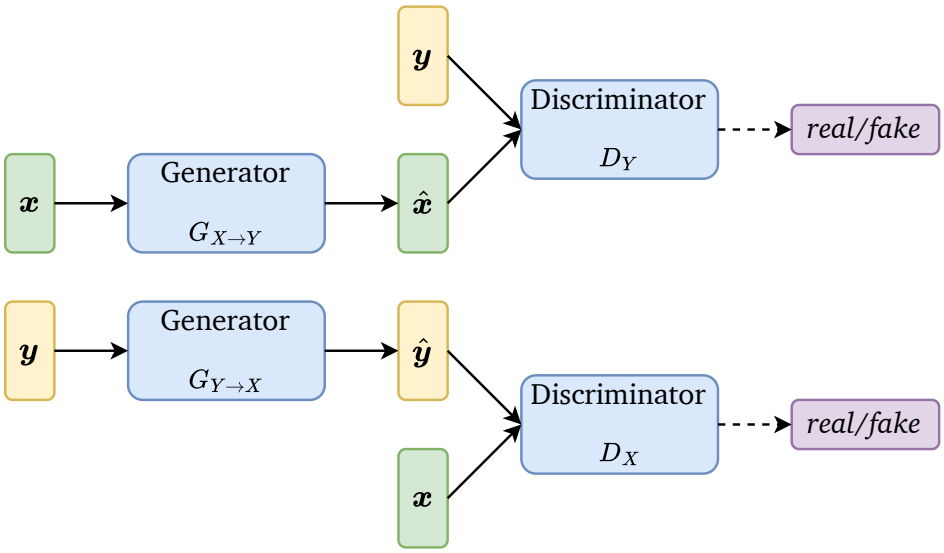
Similarly, the loss function of the backward adversarial process can be computed by using:

$$\begin{aligned}\mathcal{L}_{adv}(G_{Y\to X}, D_X) = &\mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[log\, D_X(\boldsymbol{x})] \\ &+ \mathbb{E}_{\boldsymbol{y}\sim p_{data}(\boldsymbol{y})}[log\,(1 - D_X(G_{Y\to X}(\boldsymbol{y})))]\end{aligned} \tag{4.27}$$

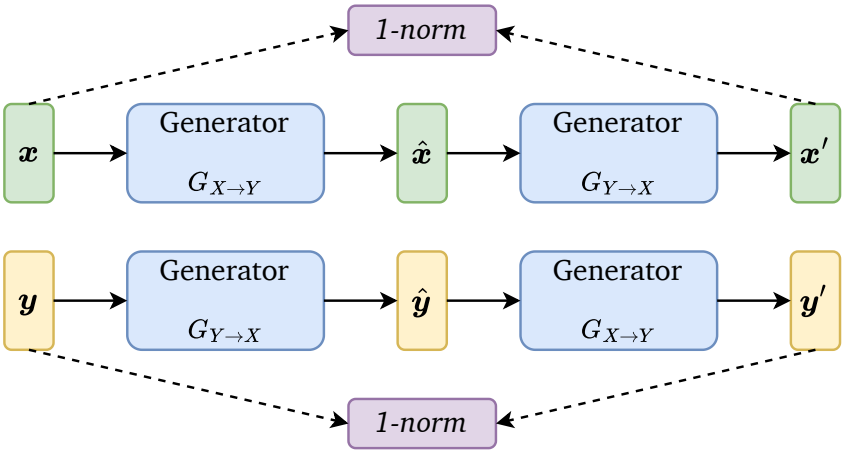
Nonetheless, the adversarial losses in both directions merely guarantee the capability of both generators to produce outputs conforming to the target or source distribution, while the content of the input might be ignored [201]. To address this,

Figure 4.10: Architectural Framework of CycleGAN

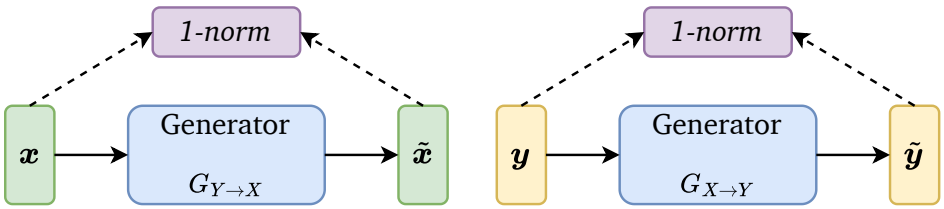
(a) Adversarial Loss



(b) Cycle Consistency Loss



(c) Identity Loss



Cycle Consistency Loss is formulated and employed to preserve the information of the input content. It achieves this by instructing the generator to map the output of the opposing generator back to the initial input, as shown in Figure 4.10b. This cyclic process is mathematically represented as:

$$\begin{aligned}\mathcal{L}_{cyc}(G_{X\to Y}, G_{Y\to X}) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\|G_{Y\to X}(G_{X\to Y}(\boldsymbol{x})) - \boldsymbol{x}\|_1] \\ + \mathbb{E}_{\boldsymbol{y}\sim p_{data}(\boldsymbol{y})}[\|G_{X\to Y}(G_{Y\to X}(\boldsymbol{y})) - \boldsymbol{y}\|_1]\end{aligned} \tag{4.28}$$

Despite the effectiveness of adversarial loss and cycle consistency loss in instructing the training of the model, an unexpected alteration in style has been observed when the source input x is fed into the opposing generator $G_{Y\to X}$ [201]. Under this circumstance, the expected output should ideally remain unchanged since the input already possesses the target style, as illustrated in Figure 4.10c. Consequently, to address this problem, Identity Loss is incorporated into CycleGAN, which is formulated as:

$$\begin{aligned}\mathcal{L}_{id}(G_{X\to Y}, G_{Y\to X}) = \mathbb{E}_{\boldsymbol{y}\sim p_{data}(\boldsymbol{y})}[\|G_{X\to Y}(\boldsymbol{y}) - \boldsymbol{y}\|_1] \\ + \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\|G_{Y\to X}(\boldsymbol{x}) - \boldsymbol{x}\|_1]\end{aligned} \tag{4.29}$$

To summarise, the loss function of the training of CycleGAN can be represented as below:

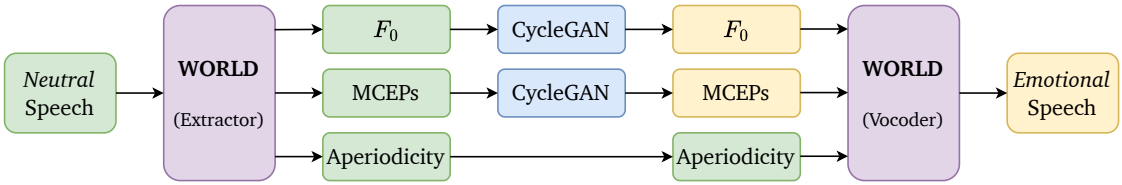
$$\begin{aligned}\mathcal{L}_{CycleGAN}(G_{X\to Y}, G_{Y\to X}, D_X, D_Y) = \mathcal{L}_{adv}(G_{X\to Y}, D_Y) \\ + \mathcal{L}_{adv}(G_{Y\to X}, D_X) \\ + \lambda_{cyc}\mathcal{L}_{cyc}(G_{X\to Y}, G_{Y\to X}) \\ + \lambda_{id}\mathcal{L}_{id}(G_{X\to Y}, G_{Y\to X})\end{aligned} \tag{4.30}$$

Here, the symbol λ represents the weighting factor applied to various losses. The similarity between image-to-image translation and VC/EVC tasks, both involving the translation of the 'style' of the input signal to another, facilitates the transfer of the utilisation of CycleGAN from the field of CV to CA.

4.3.2 Dual-CycleGAN Application

Considering the discussion on the applicable datasets in Section 3.2.3, the EmoV-DB dataset is selected as the training dataset. Additionally, as a result of the exceptional efficacy of F_0 and MCEPs in both VC and EVC studies, CWT-F_0 was selected as the prosodic feature while MCEPs were designated as the spectral feature for CycleGAN application. Recognising the different physical significance of prosody and spectrum, the requirement arose for two distinct CycleGANs—one to convert

Figure 4.11: Architectural Framework of Dual-CycleGAN EVC System



the prosodic feature and another for the spectral feature. Consequently, a model integrating two parallel CycleGANs, termed 'Dual-CycleGAN', was implemented and researched [156].

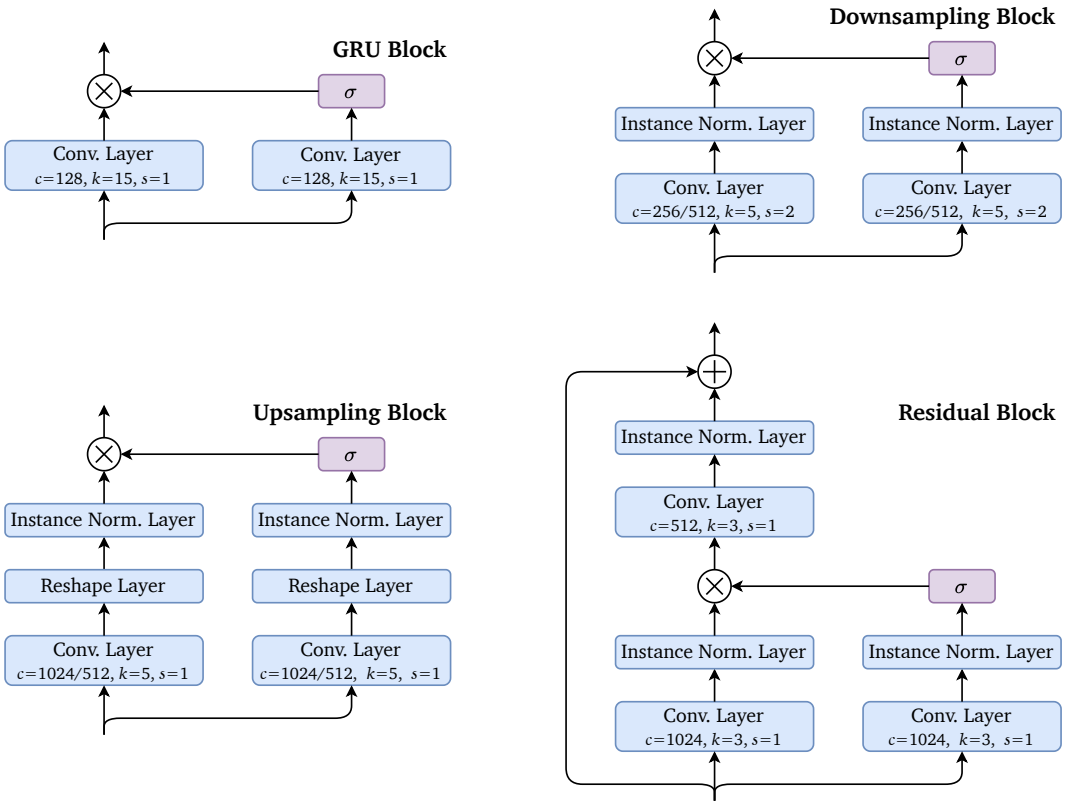
4.3.2.1 Architectural Framework

The architectural framework of the Dual-CycleGAN EVC system is illustrated in Figure 4.11. During the training phase, a pair of speech samples, consisting of source (neutral) and target (emotional) samples, undergo processing using the WORLD vocoder to extract F_0 and SSEs. Subsequently, while the SSEs from the source sample are encoded to MCEPs and then channelled into one CycleGAN—referred to as Spectral CycleGAN—for conversion, the F_0 from the source requires to be transformed through CWT before entering the other CycleGAN, which is referred to as Prosodic CycleGAN. Essentially, these two CycleGANs operate independently to convert the source CWT-F_0 and MCEPs, respectively. The backpropagation update for both CycleGANs depends on the converted source CWT-F_0 and its corresponding target, alongside the converted source MCEPs and their target counterparts.

After the training, both trained CycleGANs are employed in the inference phase. Feature extraction, including CWT and the spectral coding, and the conversion process remain the same as in the training phase. However, instead of updating models, the converted CWT-F_0 and MCEPs are leveraged to reconstruct speech by using the WORLD vocoder. It is important to note that to generate speech, the WORLD vocoder [53] requires prosodic features (F_0), spectral features (spectral envelope) and aperiodicity features (band aperiodicity, which represents the power ratio between the speech and the aperiodic component in a signal [203]). Hence, the converted CWT-F_0 requires to be transformed back to its original state using Equation 4.6 before the speech reconstruction. Similarly, the converted MCEPs need to be decoded by WORLD to obtain the converted spectral envelope. Additionally, the necessary band aperiodicity is taken directly from the source speech sample, because there is no emotional information incorporated in it [155].

Both the prosodic and spectral CycleGAN models utilise the same network architecture, as applied in [156]. The generator consists of 5 key modules, beginning with 1 gated 1D convolutional block, designed to address the gradient vanishing and explosion issues typically associated with traditional recurrent layers, while

Figure 4.12: Network Settings of All Modules in Dual-CycleGAN EVC System



also enhancing processing speed [204]. This is followed by 2 downsampling blocks, 6 residual blocks, and 2 upsampling blocks. At the generator's end, a 1D convolutional layer is applied to generate output. In contrast, the discriminator's architecture is simpler, comprising a gated 2D convolutional block, three downsampling blocks, and a fully connected layer. The detailed architectures of all modules are depicted in Figure 4.12.

4.3.2.2 Experimental Setups and Results

In this experiment, the EmoV-DB dataset was selected for training, focusing on 50 groups of speech samples performed by the speaker *Spk-Bea* due to the gender similarity. The F_0 extraction process began with a frame window duration of 5 ms, constrained within a frequency range of 71 Hz to 800 Hz. Following this, CWT was applied to the extracted F_0 using the Mexican hat wavelet function, defined in Equation 4.3, as the mother wavelet. The frame window size was maintained at 5 ms, and 10 scales were extracted with an interval of 1, resulting in a 10-dimensional prosodic feature.

For spectral feature extraction, SSEs were computed from the speech signal and F_0 using WORLD, with an Fast Fourier Transformation (FFT) size of $1,024$. The SSEs comprised 513 dimensions, which were then encoded by WORLD to produce 24-dimensional MCEPs, suitable for use in the spectral CycleGAN.

To enhance model performance, the extracted CWT-F_0 and MCEPs were normalised using the mean μ and standard deviation σ across the entire training set, with this Mean-Std Normalisation expressed as:

$$\hat{x} = \frac{x - \mu(x)}{\sigma(x)} \tag{4.31}$$

As with the approach described in Section 3.2.6, and given that the trained EVC model was used to convert synthetic speech from the Blizzard 2011 dataset rather than the EmoV-DB dataset, regular human perception evaluation was conducted instead of applying a validation set. Adam was set as the optimiser, with learning rates of 2×10^{-4} for the generator and 1×10^{-4} for the discriminator to optimise training performance.

The training results for both CycleGANs across 4 different emotional categories are presented in Figure 4.13. The loss values for all four MCEP generators converged similarly across the four EVC models, decreasing from around 30.0 to below 5.0, while the loss values of the CWT-F_0 generators reached approximately 3.0. After $4,000$ epochs, the loss curves began to flatten. Based on human perception evaluation, one spectral CycleGAN and one prosodic CycleGAN were selected from the saved models during training to convert MCEPs and CWT-F_0. The converted CWT-F_0 was then processed, and combined with the band aperiodicity of the source speech to reconstruct the speech using the WORLD vocoder. The converted speech in 4 emotional states demonstrated corresponding emotional expression while preserving linguistic information with good intelligibility.

4.3.3 Mono-CycleGAN Application

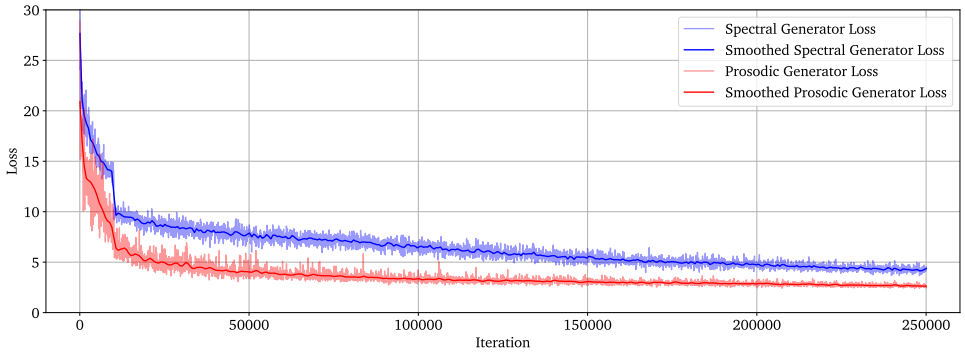
Although the Dual-CycleGAN system demonstrates decent performance on the EVC task, there is still space for improvement. Given the complexity involved in optimising two CycleGANs with different objectives, a similar system based on a single GAN model was proposed [155].

4.3.3.1 Architectural Framework

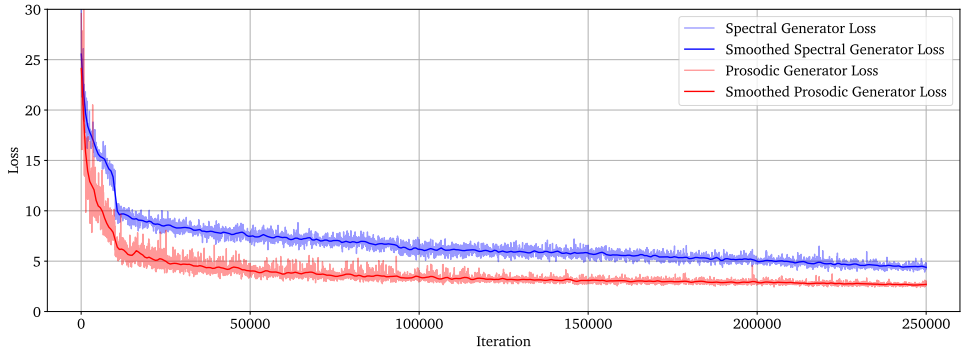
Instead of using two GANs to convert the prosodic and spectral features separately, a simpler approach called Log Gaussian Normalisation is applied for the conversion of F_0 [155], instead of applying a CycleGAN. The mathematical expression for log Gaussian normalisation is:

Figure 4.13: Loss Curves of EVC System of Dual-CycleGAN with EmoV-DB

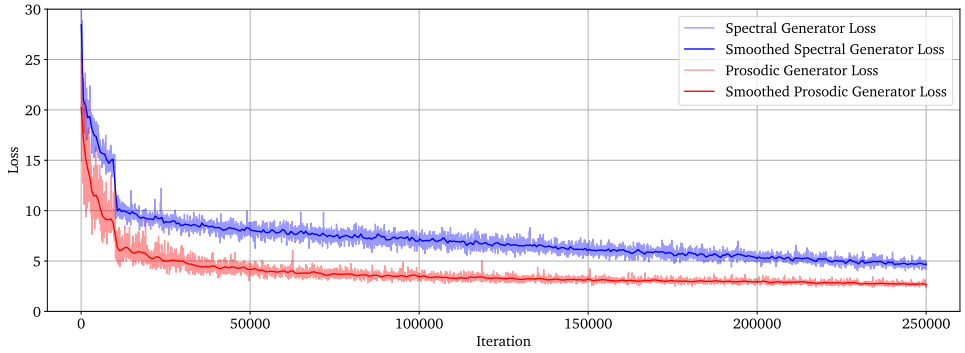
(a) *Amused*



(b) *Angry*



(c) *Disgust*



(d) *Sleepy*

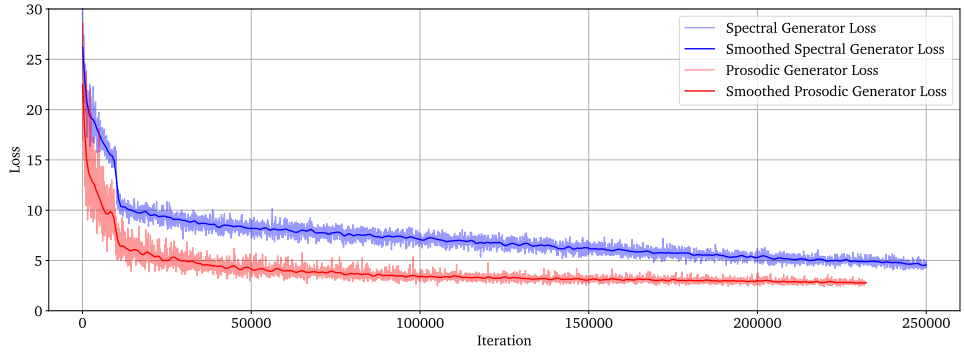
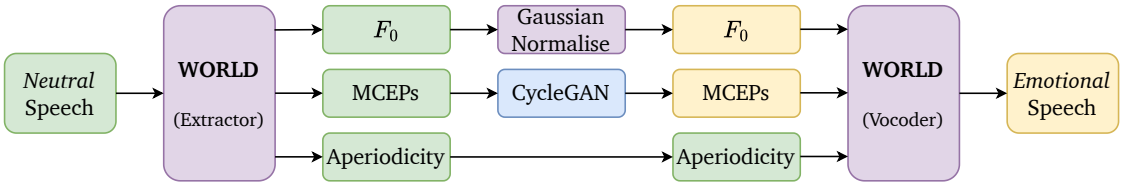


Figure 4.14: Architectural Framework of Mono-CycleGAN EVC System



$$\hat{F}_0 = exp(\frac{\sigma_t}{\sigma_s} \cdot (logF_0 - \mu_s) + \mu_t) \tag{4.32}$$

where F_0 and $\hat{F}_0$ represent the fundamental frequency of the source and converted speech. The variables μ and σ denote the mean and standard variance of f_0 across all speech samples, with subscripts s and t referring to the source and target datasets of the calculated samples, respectively.

Additionally, the proposed system employs a CycleGAN model, as introduced in Section 4.3.2, replacing the original Autoencoder GAN used in the state-of-the-art [155]. The architecture of this Mono-CycleGAN EVC system is depicted in Figure 4.14, where the CycleGAN focuses solely on converting the MCEPs of the source speech. As in previous methods, the band aperiodicity of the source speech is directly utilised by the WORLD vocoder.

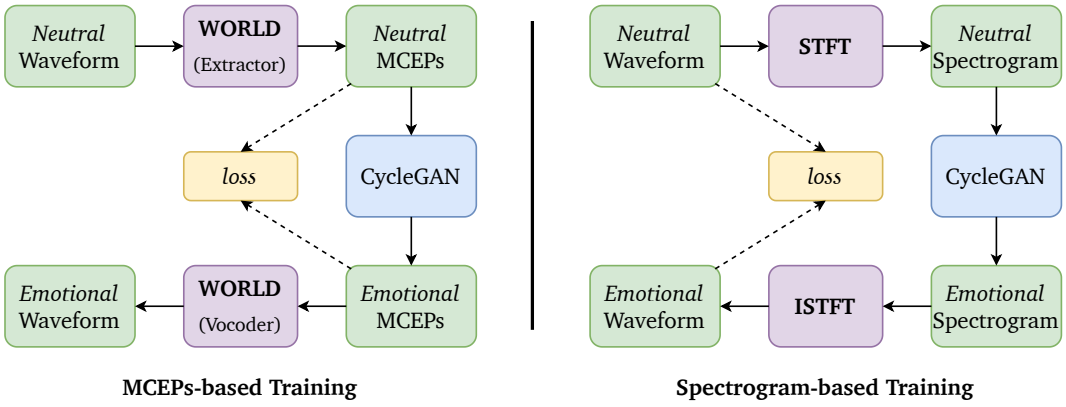
4.3.3.2 Experimental Setup and Results

Given the high similarity between the Mono-CycleGAN EVC system and the Dual-CycleGAN EVC system—both aiming to generate MCEPs—the trained models from Section 4.3.2.2 were directly applied. After reconstructing the speech using the converted MCEPs (via spectral decoding), linearly normalised F_0 and the source band aperiodicity, the Mono-CycleGAN EVC system achieved performance comparable to the Dual-CycleGAN system when converting synthetic speech produced by the neutral TTS system from Section 2.2, as evaluated by human perception. However, the model's complexity was nearly halved, resulting in reduced storage space and shorter model loading times during the inference phase. This significant advantage of the Mono-CycleGAN EVC system prompted further investigation.

4.3.4 Exploration of Improvement on CycleGAN

As demonstrated above, both the Dual-CycleGAN and the Mono-CycleGAN systems effectively preserve linguistic information. However, improvements are still needed in emotional expression and speech quality. To address these shortcomings,

Figure 4.15: Conversion of F_0 and MCEPs v.s. Conversion of Spectrogram



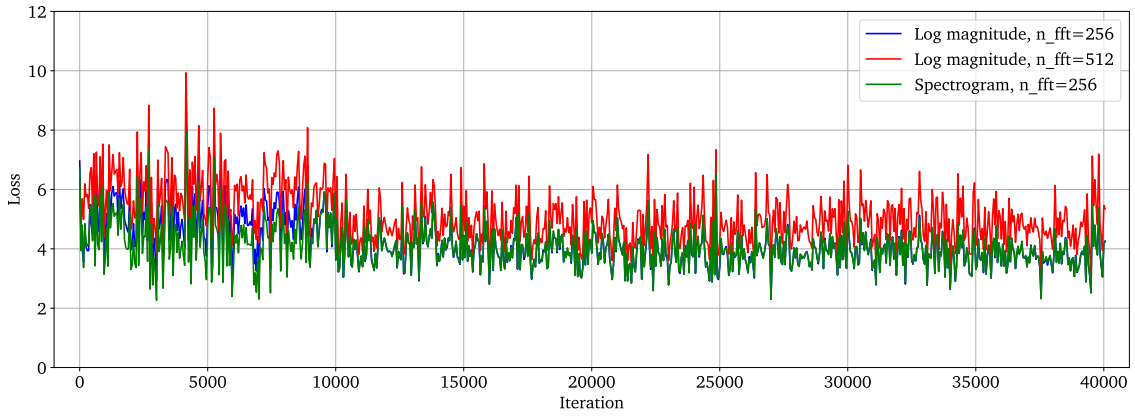
an approach that considers error accumulation throughout the process was proposed and subsequently investigated through experiments.

According to the experiments and state-of-the-art methods discussed above, CycleGANs have demonstrated significant capability in mapping between different emotional states in non-parallel training schemes. However, challenges remain in the EVC procedure. The left side of Figure 4.15 illustrates the current EVC procedure during the inference phase. The process begins with extracting MCEPs from the source speech sample, followed by feeding these MCEPs into a CycleGAN model to generate the converted MCEPs. In this figure, the log Gaussian normalisation of F_0 is omitted. Subsequently, the WORLD vocoder reconstructs the speech using the converted features. The loss functions are then calculated based on the difference between the converted and target MCEPs.

It is crucial to note that four types of errors accumulate throughout this procedure: one from the CycleGAN mapping, another from log Gaussian normalisation, and two additional errors stemming from feature extraction and speech reconstruction by the vocoder [53]. In fact, experiments have shown that the WORLD vocoder cannot perfectly reconstruct speech even when using its own extracted acoustic features. These cumulative errors adversely affect the accurate mapping capabilities of the EVC system.

To mitigate the impact of these cumulative errors, an approach to optimise the procedure was proposed. As depicted in the right part of Figure 4.15, the spectrogram was selected as the acoustic feature instead of both F_0 and MCEPs. The inverse STFT can nearly perfectly reconstruct speech using the spectrogram, and reducing the number of utilised features also decreases the unnecessary errors caused by F_0 and MCEPs. Moreover, the loss functions in the training phase are calculated using the converted and target waveforms directly—in other words, loss calculation is based on the time-domain rather than the frequency-domain. This

Figure 4.16: Loss Curves of Mono-CycleGAN with Spectral Features



modification has the advantage of making the differences between the converted and target speech more straightforward.

However, after implementing the spectrogram-based system, results showed that its performance was inferior to the baseline. Several experiments were conducted, including variations in spectrogram and log magnitude, as well as different FFT lengths (256 and 512). The results of these attempts with conversion between *Neutral* and *Amused* are illustrated in Figure 4.16. Notably, models under all three setups failed to converge during training, indicating an inability to learn the mapping between source and target speech. Additionally, human perception evaluations revealed that the converted speech had much worse quality than the baseline, with a significant loss in intelligibility. The primary reason for the failure of spectrogram is that its 'perfectly reconstructable' nature also means it contains too much information for effective EVC, especially in non-parallel training. Therefore, retaining both prosodic and spectral features, such as F_0 and MCEPs, remains a prudent choice despite the complexity and potential for errors in the procedure.

4.4 Application: VAE-GAN

In the proposed ETTS system, neutral speech is first generated by a neutral TTS module and then processed through an EVC system. Since the training datasets used by the neutral TTS model and the EVC model involve different speakers, the EVC model must handle unseen speakers during the inference phase. As discussed in Section 4.3.4, the CycleGAN-based EVC system struggles to effectively convert the voices of unseen speakers. To address this issue, an alternative neural network architecture known as VAW-GAN, which is claimed to have the capability to process unseen speakers [151], was considered. However, VAW-GAN's reliance on its approximated metric, VAEs with Wasserstein Distance, often requires arbitrary

choices and may not lead to the desired Gaussian distribution [205]. Therefore, inspired by another study on non-parallel GAN-based EVC [146], VAE-GAN was implemented on the EmoV-DB dataset to explore its performance with unseen speakers.

4.4.1 VAE-GAN

As introduced in Section 4.2.4.1, VAEs have the ability to generate content due to their probabilistic nature. VAEs benefit from the generator-discriminator architecture of GANs, forming a combined network known as VAE-GAN, where the decoder of the VAE also functions as the generator in the GAN.

A typical VAE-GAN architecture involves three models: an Encoder, a Generator, and a Discriminator. The encoder processes the input data to produce the mean and log variance parameters, which are used to construct a Gaussian distribution. These parameters define a latent space from which a representation can be sampled. The generator then takes this latent representation to estimate a new data representation [8]. This forms the classic VAE model. In a VAE-GAN, a discriminator is added to evaluate a real/fake decision of the generated representation, guiding the generator to produce more realistic samples, similar to its role in a regular GAN.

The encoder's training is guided by two loss functions: Kullback-Leibler Divergence (KLD) and Reconstruction loss. KLD measures the distance between two probability distributions P and Q, denoted as $D_{KL}(P \,||\, Q)$ [206]. In VAEs and VAE-GANs, it is used to measure the distance between the latent distribution $q(z|x)$ produced by the encoder and the prior distribution $p(z)$, denoted as $D_{KL}(q(z|x) \,||\, p(z))$.

A challenge arises because the prior distribution of the latent space is unknown and unobtainable. However, this issue can be mitigated if the decoder/generator of the VAE/VAE-GAN has sufficient expressivity for reconstruction, allowing the shape of the prior distribution to be arbitrary [207]. For computational simplicity, a standard Gaussian distribution $\mathcal{N}(0,1)$ is commonly used as the prior distribution. The Kullback-Leibler loss function in VAEs/VAE-GANs is therefore defined as:

$$\mathcal{L}_{KL}(\mu, \sigma^2) = D_{KL}(\mathcal{N}(\mu, \sigma^2) \,||\, \mathcal{N}(0,1)) \tag{4.33}$$

where μ and σ denote the mean and standard deviation obtained from the encoder's output. To simplify the calculation involving $\mathcal{N}(0,1)$:

$$D_{KL}(\mathcal{N}(\mu, \sigma^2) \,||\, \mathcal{N}(0,1)) = \frac{1}{2}(\mu^2 + \sigma^2 - log(\sigma^2) - 1) \tag{4.34}$$

Another common measure used in VAE architectures to replace KLD is the Wasserstein Distance. This combination forms what is known as VAW-GAN, which

has been applied and explored in non-parallel VC [144], EVC [151] and singing voice conversion [208].

Returning to the classical VAE/VAE-GAN, the reconstruction loss involves minimising the negative log-likelihood, expressed as:

$$\mathcal{L}_{recon} = -\mathbb{E}_{q_\phi(z|x)}(log\, p_\theta(x|z)) \tag{4.35}$$

where $q_\phi(z|x)$ and $p_\theta(x|z)$ are the posterior probability distributions from Equations 4.23 and 4.24. The encoder's optimisation is achieved by summing these two loss functions:

$$\mathcal{L}_E = \lambda_{KL}\mathcal{L}_{KL} + \lambda_{recon}\mathcal{L}_{recon} \tag{4.36}$$

where λ_{KL} and λ_{recon} are weights to balance the training.

The generator and discriminator losses in VAE-GAN follow those of a regular GAN, where the generator aims to produce data that is both close to the target and able to fool the discriminator. The generator-discriminator interplay is captured in the loss function:

$$\mathcal{L}_G = \lambda_{recon}\mathcal{L}_{recon} + \lambda_{adv}\mathcal{L}_{adv} \tag{4.37}$$

where λ_{recon} and λ_{adv} are the respective weights, and the adversarial loss $\mathcal{L}_{adv}$ is defined as:

$$\mathcal{L}_{adv} = -\mathbb{E}_{z\sim p(z)}[log\, D(G(z))] \tag{4.38}$$

where z represents the latent representation sampled from the latent distribution $p(z)$.

The discriminator's role is to distinguish between real data x and generated data $G(z)$. The discriminator loss function $\mathcal{L}_D$ is expressed as:

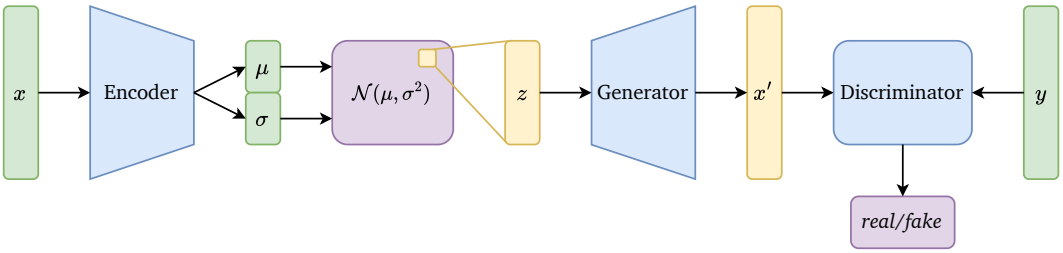
$$\mathcal{L}_D = -\mathbb{E}_{x\sim p_{data}(x)}[log\, D(x)] - \mathbb{E}_{z\sim p(z)}[log\,(1 - D(G(z)))] \tag{4.39}$$

In summary, the overall loss function of VAE-GAN is the sum of the loss values of all three components:

$$\mathcal{L}_{VAE-GAN}(E, G, D) = \mathcal{L}_E + \mathcal{L}_G + \mathcal{L}_D \tag{4.40}$$

The architectural framework of VAE-GAN is illustrated in Figure 4.17. VAE-GAN has demonstrated its capabilities in various tasks, including VC [209], EVC [146, 176], music generation [210] and other generative tasks [211].

Figure 4.17: Architectural Framework of VAE-GAN



4.4.2 Dual-VAE-GAN Application

Given the excellent performance of F_0 and MCEPs, and the fact that spectral features like spectrograms are not suitable to be used as the sole feature in EVC, the first proposed EVC system based on VAE-GAN adopts a dual-model approach. In this system, each model is responsible for converting the prosodic and spectral features, respectively. This design has proven to be highly effective in handling speaker-independent emotional representation, owing to its encoder-decoder architecture [151].

4.4.2.1 Architectural Framework

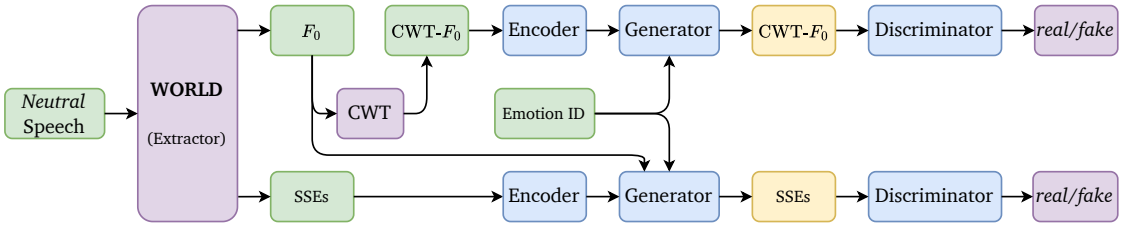
The training process of the Dual-VAE-GAN EVC system is illustrated in Figure 4.18. The conversion procedure is highly similar to that of the Dual-CycleGAN EVC system, with the training phase comprising the following steps:

1. **Feature Extraction**. The WORLD vocoder is used to extract prosodic features (F_0), spectral features (SSEs) and aperiodicity features (band aperiodicity). However, the aperiodicity feature is not used during the training phase.

2. **Prosodic Feature Conversion**. The extracted F_0 is first transformed using CWT to obtain CWT-F_0, which is then fed into the encoder of the prosodic VAE-GAN model. The output, enriched with emotional information, is passed through the prosodic generator to generate the converted prosodic feature.

3. **Spectral Feature Conversion**. The process of converting spectral features follows a similar approach to that used for F_0, with two key differences. Firstly, SSEs can be directly fed into the spectral encoder without any additional transformation. Secondly, since the spectral feature incorporates prosodic information and relies on it to some extent [151], the conversion of spectral features benefits from using F_0 as an additional input to the spectral generator.

Figure 4.18: Training Process of Dual-VAE-GAN EVC System



4. **Loss Calculation**. The converted prosodic and spectral features, along with their ground truth counterparts, are used to compute the loss values for backpropagation and model updates. Additionally, two discriminators are employed to predict the real/fake decision of both converted features, contributing to the adversarial loss calculation, which in turn enhances the generative performance of the two generators.

In the inference phase, the process can be divided into four similar steps, as depicted in Figure 4.19:

1. **Feature Extraction**. Prosodic, spectral and aperiodicity features are extracted using WORLD.

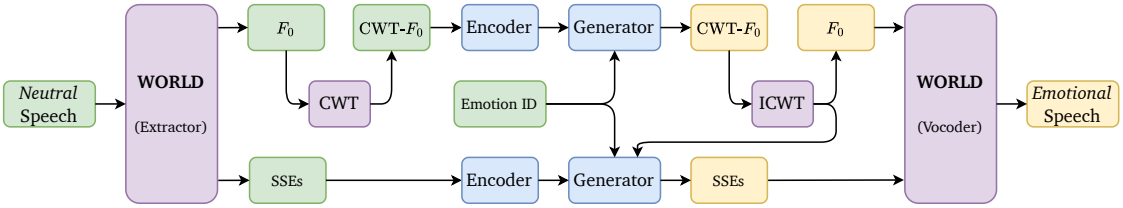
2. **Prosodic Feature Conversion**. This step mirrors the training phase, using the trained generator to convert the CWT-F_0 of the input speech. However, to synthesise the speech using the converted features, an inverse CWT is required to obtain the converted F_0.

3. **Spectral Feature Conversion**. Unlike in the training phase, the converted F_0 (obtained by applying the inverse CWT to the output of the prosodic generator) is used as input to the spectral generator instead of the original F_0.

4. **Speech Synthesis**. Finally, the converted F_0, SSEs and the source speech's band aperiodicity are combined using the WORLD vocoder to reconstruct the converted speech.

The Dual-VAE-GAN EVC system follows the network settings of [151]. Apart from the differences in feature dimensions, as seen in the Dual-CycleGAN EVC system discussed in Section 4.3.2.1, the Dual-VAE-GAN system introduces another difference: the spectral VAE-GAN generator has an extra input compared to the prosodic VAE-GAN. This extra input is the ground truth F_0 during the training phase, while the converted F_0 is used during the inference phase. Nonetheless, both VAE-GANs employ similar network modules and blocks, as illustrated in Figure 4.20, with this slight difference. The encoder comprises 5 2D convolutional layers, followed

Figure 4.19: Inference Process of Dual-VAE-GAN EVC System



by 2 separate fully-connected layers that generate 2 vectors from the flattened output. The mean and log variance vectors obtained from the fully-connected layers are then used to define a Gaussian distribution.

The generator takes a sample from the latent Gaussian distribution, along with the Emotion ID and F_0 (in the case of the spectral VAE-GAN), as input. The Emotion ID and F_0 are first processed through an embedding layer. The latent sample and the embeddings of the Emotion ID (and F_0) are then processed by 2 (or 3) fully-connected layers, respectively. The sum of these outputs is fed into another fully-connected layer. Finally, 4 2D transposed convolutional layers are used to generate the converted feature. The discriminator, on the other hand, consists of 3 2D convolutional layers to process the input feature, followed by a fully-connected layer that provides the real/fake decision for both the ground truth and the converted samples.

4.4.2.2 Experimental Setup and Results

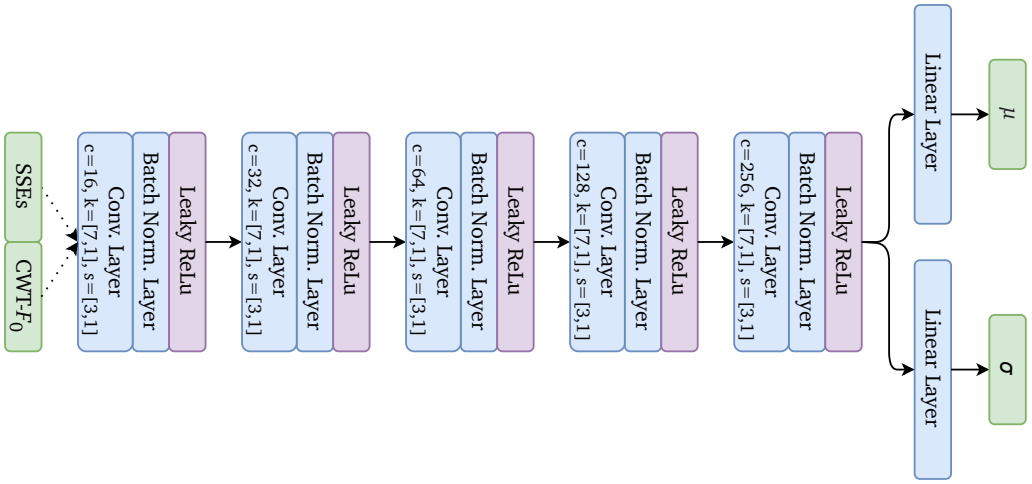
The VAE-GANs were trained using the EmoV-DB dataset, similar to the CycleGAN-based EVC systems, but with two key differences. Firstly, speech samples from two female speakers, *Spk-Je* and *Spk-Bea*, were selected, instead of only using samples from *Spk-Bea*. Secondly, while 10 samples from each speaker in each emotional category were set aside for human evaluation, all other samples were included in the training set. This resulted in a training set comprising 764 *Neutral* samples, 508 *Amused* samples, 790 *Angry* samples, 512 *Disgust* samples, 953 *Sleepy* samples, totalling 2,574 samples.

Given that the SSEs lose information when encoded to MCEPs, the SSEs were chosen as the spectral feature in the Dual-VAE-GAN system. The same process of extracting a 513-dimensional SSEs as described in Section 4.3.2.2 were applied, with the exception of the spectral encoding procedure. Subsequently, the SSEs were normalised using the common logarithm to reduce the influence of high variance. Additionally, Min-Max Normalisation was applied across the dataset, defined as:

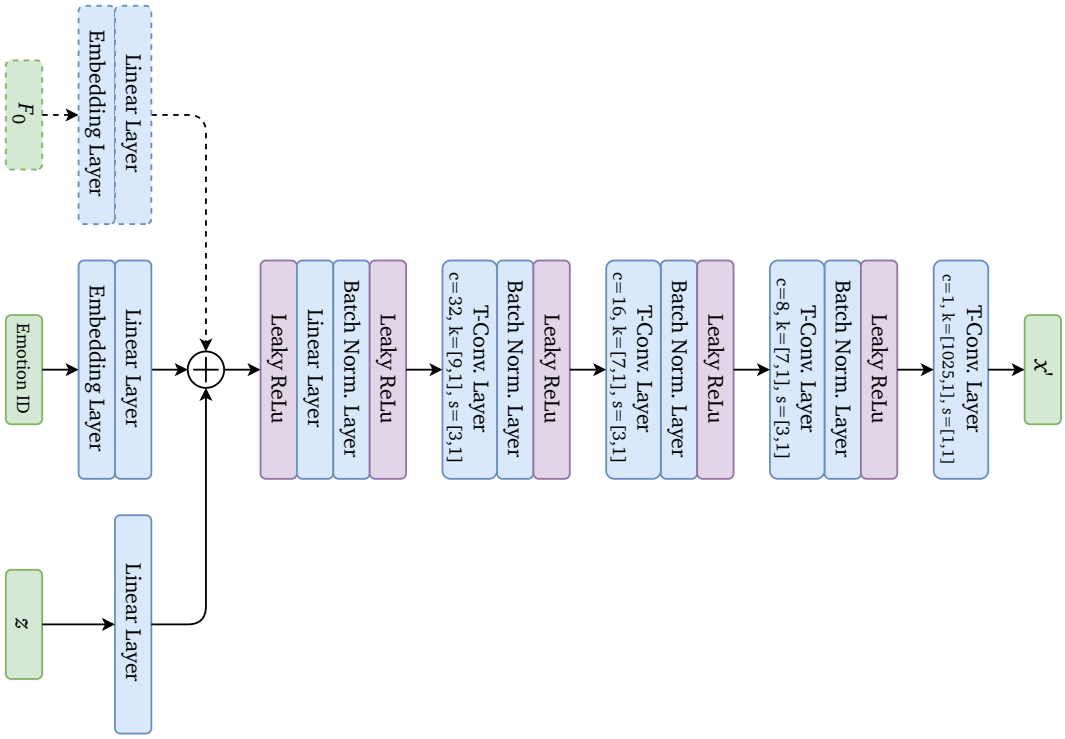
$$\hat{x} = \frac{x - min(x)}{max(x) - min(x)} \tag{4.41}$$

Figure 4.20: Network Settings of All Modules in Dual-VAE-GAN EVC System

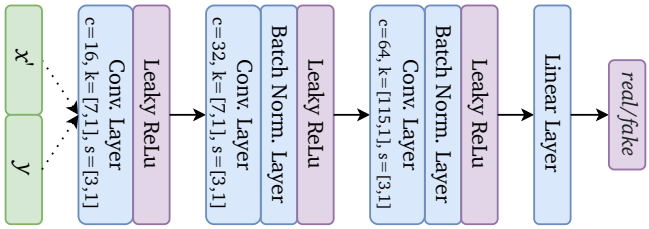
(a) Encoder



(b) Generator



(c) Discriminator



To minimise the impact of outliers, the 0.5th and 99.5th percentiles were used in place of the minimum and maximum in Equation 4.41. Meanwhile, the CWT-F_0 was normalised using Mean-Std Normalisation.

The model was optimised using RMSProp with a learning rate of 1×10^{-4} and a decay rate of 0.9. Initially, the VAE (encoder and generator) was pre-trained for 15 epochs to stabilise the training process and provide a solid initial representation for both the encoder and generator. Following this, the discriminator was introduced into the training process for an additional 45 epochs to enhance the generator's performance.

Since the three loss values were not of the same order of magnitude, a substantial adversarial weight $\lambda_{adv} = 5 \times 10^3$ was applied. As depicted in Figure 4.21, the loss curves of the three modules in the system are shown. Notably, there is a significant change in the loss values of the discriminator and generator around $11,000$ iterations, which marks the transition between the grey and white areas. Additionally, the loss curve of the discriminator exhibited severe and random oscillations before approximately $11,000$ iterations. These phenomena can be attributed to the discriminator not being trained until the 16th epoch, causing the absence of the adversarial loss $\mathcal{L}_{adv}$ in the parameter updates to influence the losses of both the generator and discriminator.

The initial phases of the encoder training, particularly before the participation of the discriminator, show a rapid decrease in loss values, showing the encoder's quick adaptation to the fundamental structures within the speech data. This phase is crucial for establishing a baseline understanding of the inherent patterns of the speech in the different emotional states (*Amused*, *Angry*, *Disgust* and *Sleepy*) and the different representations (prosodic and spectral).

As training progresses, the loss curves exhibit gradual stabilisation, with less obvious decreases in loss values. This stabilisation reflects the excellent performance of the encoder in learning and its capacity to balance objectives of minimising both the Kullback-Leibler loss $\mathcal{L}_{KL}$ and the reconstruction loss $\mathcal{L}_{recon}$. The KLD component encourages the encoder to produce a latent representation that approximates a standard Gaussian distribution, promoting the generative aspect of the model, while the reconstruction component ensures that the encoded features can be effectively reconstructed back into meaningful speech data.

As for the generator, during the initial grey area, where the discriminator does not participate in training, the loss curves for all eight models exhibit relatively stable and convergent trends. This phase is critical for the generator as it focuses on minimising the reconstruction loss $\mathcal{L}_{recon}$, without the adversarial component coming into training. The convergence observed in this phase indicates the generator's capability to learn and reproduce the input features accurately, laying a foundational understanding of the data without the pressure of fooling the discriminator.

Transitioning into the white area, post $11,000$ iterations, the introduction of the discriminator into the training process marks a significant shift in the generator's loss

behaviour. The adversarial loss component, magnified by a large weight, injects a competitive dynamic into the training, aiming to enhance the generator's ability to produce realistic outputs that can fool the prediction by the discriminator. The result is a notable value increase in loss oscillations, reflecting the generator's adjustments in response to the feedback of the discriminator. These oscillations signify the learning struggle and adaptation, as the generator endeavours to optimise both reconstruction and adversarial objectives.

The discriminator was not active in the initial 15 epochs, roughly corresponding to the first 11,000 iterations and a grey area in the figure. As training progresses beyond the grey area, the loss curves for all eight models begin to exhibit a more stable and convergent behaviour. This stability suggests that the discriminator is effectively learning to distinguish between the ground truth and the generated emotional speech features as it receives training updates.

Among all three figures, the distinction between the prosodic and spectral models in terms of their loss curves indicates different learning dynamics, due to the inherent differences in the complexity and characteristics of the prosodic and spectral features. Besides, the difference among four emotional states in the same representation reflects the inherent complexity of each emotional state. Following the human perception tests, it was observed that all converted speech samples exhibited excellent intelligibility and clear emotional expressions, while the speech quality did not meet the desired level of satisfaction. However, due to the considerable expense associated with systematic subjective evaluations, particularly in terms of time, a subjective evaluation was specifically applied to the improved Dual-VAE-GAN EVC system, as detailed in Section 4.4.3.3.

4.4.3 Optimisation of VAE-GAN

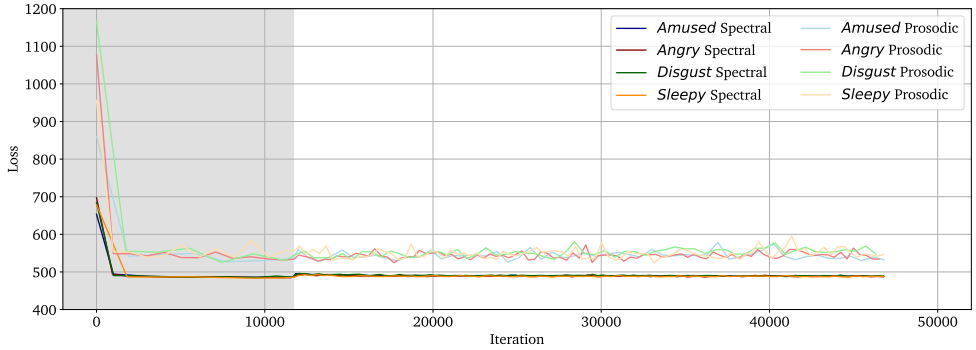
The experiment described above demonstrates that the speech quality of the converted speech produced by the Dual-VAE-GAN EVC system is not sufficiently satisfactory. Consequently, it is crucial to design and investigate effective and practical improvements to enhance the system's performance.

4.4.3.1 Optimisation of Process

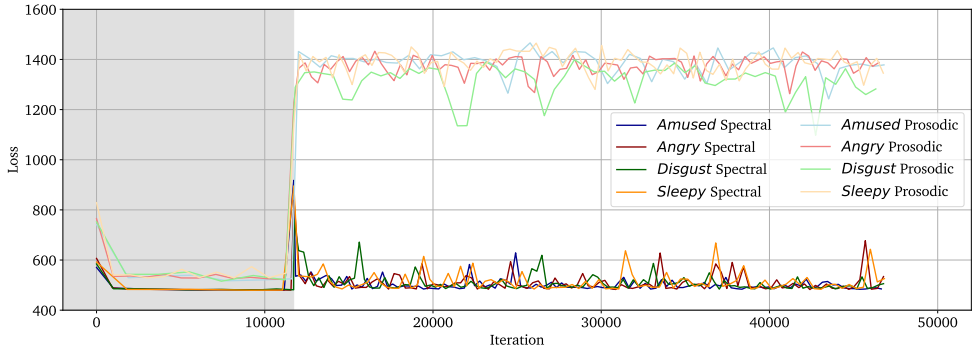
The first approach to improving the Dual-VAE-GAN EVC system draws inspiration from the existing EVC system's process. As outlined in Section 2.2 and depicted in Figure 4.3, the current ETTS system—whether based on CycleGAN or VAE-GAN—incorporates a speech-to-speech EVC module. In the upper part of Figure 4.22, the input text is processed by Tacotron 2, generating 18-dimensional BFCCs and 2-dimensional pitch features. LPCNet then synthesises neutral speech using these 20-dimensional features. The synthetic speech is subsequently processed by a feature extractor to obtain F_0 and SSEs (or MCEPs in CycleGANs), which

Figure 4.21: Loss Curves of EVC System of Dual-VAE-GAN with EmoV-DB

(a) Encoder Loss $\mathcal{L}_E$



(b) Generator Loss $\mathcal{L}_G$



(c) Discriminator Loss $\mathcal{L}_D$

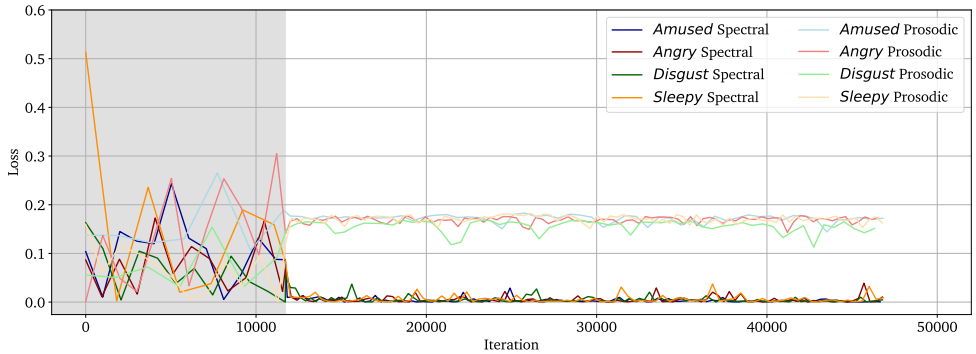
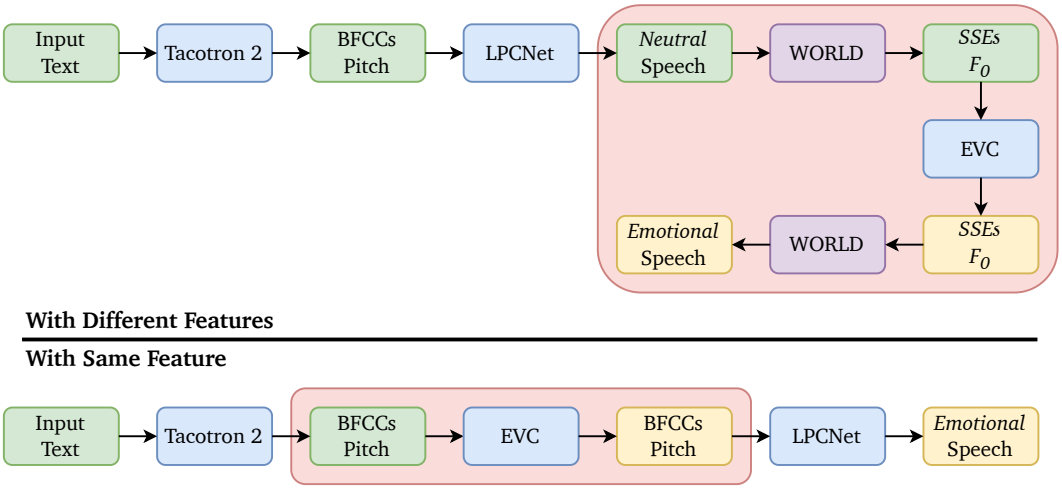


Figure 4.22: Comparison between ETTS Systems with the Different Features



the EVC model converts to emotional features. Finally, emotional speech is reconstructed using these converted features.

However, the current ETTS process involves two feature extraction stages and two speech synthesis stages, due to the differing feature sets used in the TTS and EVC modules. This procedure can be optimised for both performance and synthesis speed by unifying the feature sets used in both modules. As illustrated in the lower part of Figure 4.22, the new approach involves directly converting the extracted features to emotional BFCCs and pitch features within the EVC module. LPCNet then reconstructs the emotional speech using these converted features. This procedure reduces information loss and accelerates synthesis.

Three different frameworks for this optimisation are proposed, as shown in Figure 4.23. The first approach involves converting a concatenation of BFCCs and pitch features using a single VAE-GAN model, which considers the relationship between prosodic and spectral features. The second approach employs two separate VAE-GAN models to convert BFCCs and pitch features independently. The third approach is inspired by the Mono-CycleGAN EVC model described in Section 4.3.3, where BFCCs are converted using a VAE-GAN model, while a linear transformation is applied to pitch features. This method leverages the physical similarities between prosodic features, as well as between spectral features.

However, experiments with all three approaches revealed that the speech quality and intelligibility of the converted speech were compromised. The primary reason for this is that LPCNet is an autoregressive model, meaning that the content of each timestep is generated based on the prediction from the previous 16 timesteps [62]. This characteristic makes the current prediction heavily dependent on the accuracy of previous predictions, as errors from earlier timesteps accumulate in future

Figure 4.23: Three Approaches for Process Optimising on VAE-GAN EVC Model

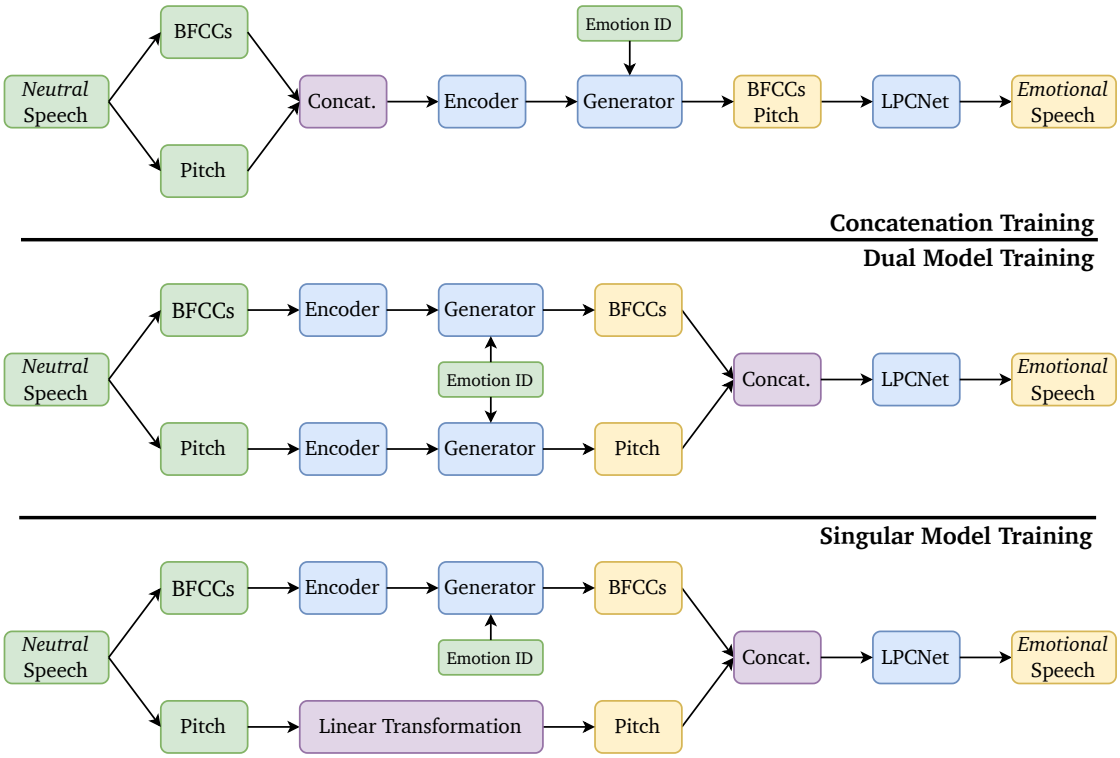
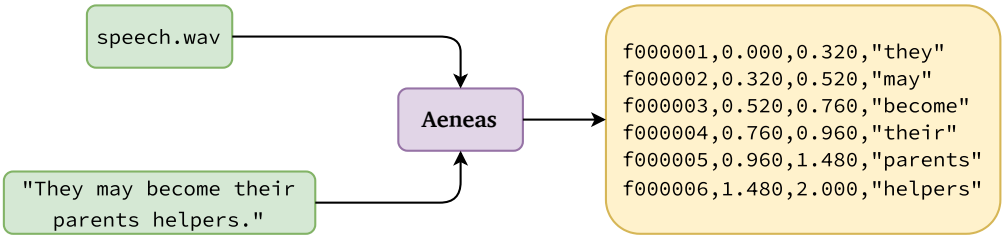


Figure 4.24: Schematic Diagram of Forced Alignment with Aeneas



predictions. As a result, mapping the neutral and emotional features in a way that LPCNet can effectively utilise proves challenging and may require a significantly large dataset for training.

4.4.3.2 Improvement of Performance

Given the similarities between VC and EVC tasks, exploring approaches that have proven effective in non-parallel VC could offer valuable insights for EVC research. One promising approach leverages the text input available in cascade ETTS systems, which has shown to be effective in VC by incorporating linguistic information.

Phonetic Posteriorgrams (PPGs) are 2-dimensional representations that display the posterior probabilities of specific phonetic classes over time [212]. PPGs have been widely studied across various domains, including VC [129, 166], accent conversion [213, 214] and singing voice conversion [215, 216], due to their ability to enhance model performance by incorporating linguistic information. For example, in one VC study, a pre-trained ASR model was used to predict PPGs for both the encoder and decoder of a VAE model, ensuring speaker-independent phonetic content [129].

In the context of a cascade ETTS system, the input is already text-based, providing direct access to linguistic information, rendering the use of an ASR model unnecessary. A more suitable technique to leverage this linguistic information is Forced Alignment. Forced alignment uses input speech and its corresponding text transcription—obtained either from ground truth or any ASR system—to align them, producing an output similkear to PPGs but with the actual words included instead of posterior probabilities.

For this experiment, Aeneas [217] was chosen as the toolkit to align the speech signal with the text transcription, as illustrated in Figure 4.24. In this figure, the output columns represent the ID, start timestamp, end timestamp and the aligned word. By applying Aeneas for alignment, one can extract linguistic information, which is then fed into the encoder along with acoustic features, according to the distribution of the pronunciation. This integration of precise linguistic information aims to enhance the performance of the EVC model.

4.4.3.3 Objective and Subjective Evaluation

The implementation of the improvement above led to both objective and subjective evaluations of the proposed EVC system. Initially, the system's real-time performance was assessed, achieving an RTF of 2.47. Additionally, three objective evaluation metrics, as introduced in Section 3.1.4, were selected to compare the proposed EVC system against the baseline, namely MCD, Log-Spectral Distortion (LSD) and RMSE of F_0. Ten different *Neutral* speech samples from the EmoV-DB dataset were chosen to be converted by both EVC models, with the converted speech samples in all 4 emotional states, including *Amused*, *Angry*, *Disgust* and *Sleepy*, being evaluated. Firstly, to compare speech samples of differing lengths, DTW was applied to all converted samples. Following this, the three metrics were calculated between the ground truth samples and the converted speech samples by the baseline, and then the same calculations were performed for the proposed EVC system and the ground truth. Thus, the comparison between the proposed and baseline EVC systems is presented in Table 4.1.

The results indicate that the proposed EVC system outperforms the baseline in *Amused* and *Sleepy*, with *Sleepy* achieving the best performance, which includes an MCD of 6.23 (baseline: 6.26), an LSD of 5.85 (baseline: 5.98) and an F_0-RMSE of 72.85 (baseline: 75.57). However, the proposed EVC system does not perform as well in *Angry* and *Disgust* under these three evaluation criteria.

Considering that the objective of EVC is not to generate speech identical to the ground truth emotional speech, the objective evaluation results above can only serve as a useful reference for model performance. Hence, subjective evaluation, based on human perception, is more convincing and necessary.

The subjective evaluation encompassed both the neutral TTS system introduced in Section 2.2 and the proposed EVC system. The entire evaluation setup involved 14 participants affiliated with the University of Augsburg or the Munich Research Centre of Huawei. Eight different sentences from the evaluation set of the EmoV-DB dataset were randomly selected, and the corresponding neutral and four emotional speech samples were subsequently chosen. After obtaining the ground truth samples, the synthetic neutral speech samples and the converted emotional speech samples derived from them were also prepared. As a result, a total of $8 \times [(1 + 4) + (1 + 4)] = 80$ samples were included.

To conduct the subjective evaluation, a questionnaire was designed, beginning with 5 self-evaluation questions, including:

1. I am an empathic person, e.g. I get sad or angry during dramatic scenes in movies.

2. I can easily hear and identify how a person feels in a phone conversation. I don't need to see them to recognise their emotions.

Table 4.1: Objective Evaluation Results of EVC System with VAE-GAN

(a) Amused

Model	**MCD**	**LSD**	**F_0-RMSE**
Baseline	6.86 ± 0.80	6.096 ± 2.19	$\mathbf{100.54 \pm 11.32}$
Proposed	$\mathbf{6.80 \pm 0.79}$	$\mathbf{5.71 \pm 2.07}$	100.91 ± 10.61

(b) Angry

Model	**MCD**	**LSD**	**F_0-RMSE**
Baseline	$\mathbf{6.97 \pm 0.75}$	$\mathbf{5.32 \pm 1.63}$	$\mathbf{101.37 \pm 9.03}$
Proposed	7.39 ± 0.86	6.11 ± 1.54	103.14 ± 11.16

(c) Disgust

Model	**MCD**	**LSD**	**F_0-RMSE**
Baseline	$\mathbf{5.29 \pm 0.52}$	$\mathbf{3.58 \pm 1.22}$	$\mathbf{76.09 \pm 7.56}$
Proposed	5.35 ± 0.53	4.50 ± 1.87	79.17 ± 7.21

(d) Sleepy

Model	**MCD**	**LSD**	**F_0-RMSE**
Baseline	6.26 ± 0.42	5.98 ± 2.92	75.57 ± 9.10
Proposed	$\mathbf{6.23 \pm 0.37}$	$\mathbf{5.85 \pm 2.94}$	$\mathbf{72.85 \pm 9.10}$

MCD: Mel-Cepstral Distortion
LSD: Log-Spectral Distortion
F_0-RMSE: Root Mean Square Error of F_0

3. I am musically trained, e.g. I can play a musical instrument.

4. High audio quality in media is important to me, e.g. I own high-end speakers.

5. I am good at predicting other people's behaviour, e.g. I know how my actions will make others feel.

All questions were followed by five options: *Strongly disagree*, *Disagree*, *Neutral*, *Agree* and *Strongly agree*. The objective of these questions was to allow participants to self-evaluate their performance in emotional perception, speech quality tolerance and their speech-related experience.

Following the self-evaluation, all 80 samples were presented to the participants one by one, accompanied by five questions with specific evaluation criteria. Participants were required to answer these questions based on the sample they had just heard. These questions include:

1. What emotion do you think of the voice expressed?

   1 - Neutral, 2 - Amused, 3 - Angry,

   4 - Disgust, 5 - Sleepy

2. If NOT neutral, what do you think of the intensity?

   1 - Very weak, 2 - Weak, 3 - Moderate,

   4 - Strong, 5 - Very strong

3. How close to human would you rate the voice speaking?

   1 - Not at all, 2 - A little bit close, 3 - Close,

   4 - Very close, 5 - Extremely close

4. Do you think the voice is clear and understandable?

   1 - Strongly disagree, 2 - Disagree, 3 - Neutral,

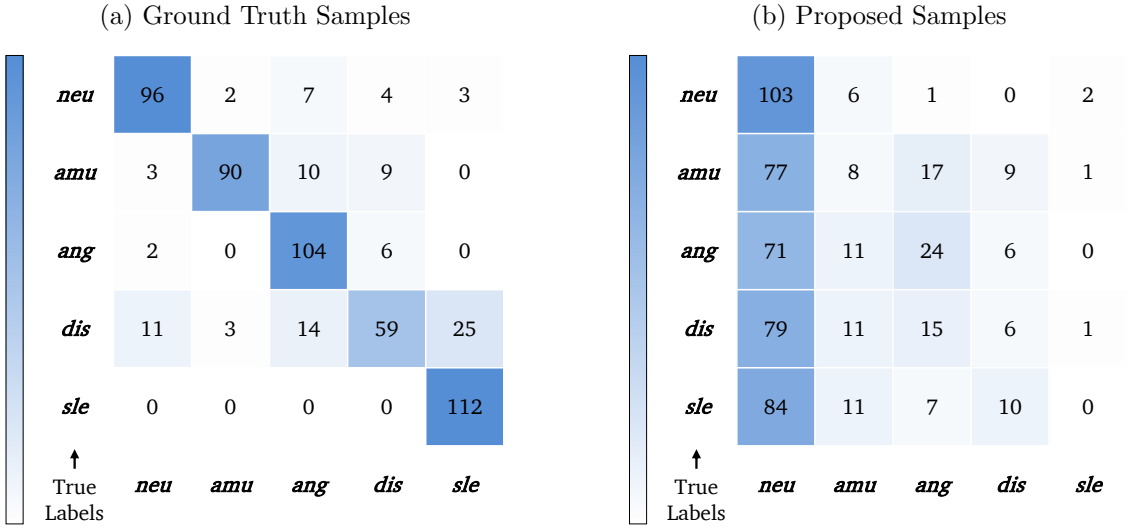
   4 - Agree, 5 - Strongly agree

5. How much do you like the voice speaking?

   1 - Not at all, 2 - Hardly, 3 - Moderately,

   4 - Greatly, 5 - Extremely

Figure 4.25: Confusion Matrix of Subjective Emotion Recognition Results



These five questions were designed based on different criteria. The first question focused on subjective SER, where participants were asked to select the emotion they perceived. The subsequent questions addressed emotional intensity, naturalness, speech quality and likeability, each using a MOS test, with a range from 1 to 5 and an interval of 1. The order of sample presentation was structured as follows: all 10 samples sharing the same linguistic content were grouped, with the order of these 8 groups decided randomly. Within each group, the sequence of the ground truth and the synthetic samples expressing different emotional states was entirely randomised. This setup allowed participants to continuously listen to and evaluate a group of samples with identical linguistic content, while the random order within each group prevented habitual decisions.

The subjective evaluation of the neutral TTS system has been detailed in Section 2.2.6, but its results are included here for completeness. Initially, the SER results of all ground truth samples are presented as a confusion matrix in Figure 4.25a as a baseline. The emotional expression of the ground truth samples from the EmoV-DB dataset is clearly perceptible, as indicated by the diagonal line from upper left to lower right in the figure. The best performance was observed in the *Sleepy* emotion, where all 112 samples were correctly classified. However, the *Disgust* emotion had the lowest performance, with an accuracy of $59 \,/\, (8 \times 14) = 52.7\,\%$.

Figure 4.25b illustrates the SER results of the proposed neutral TTS and EVC systems. Overall, the neutral TTS system performs well, as discussed in Section 2.2.6, while the proposed EVC system does not fare as well. Notably, none of the converted *Sleepy* samples were correctly recognised. However, to differentiate whether the emotional expression is incorrect or merely weak, the information within the first column from the left—which indicates how many synthetic samples were

classified as *Neutral*—is useful. If an emotional sample is classified correctly or as *Neutral*, but not as another emotional state, it can be considered 'acceptable'. According to this criterion, the acceptable rates for *Amused*, *Angry*, *Disgust* and *Sleepy* are 75.9 %, 84.8 %, 75.9 % and 75.0 %, respectively.

Given that most of the converted samples were not classified correctly in terms of emotional expression, the MOS test results on intensity are not particularly insightful. Therefore, an analysis of naturalness, speech quality and likeability is more pertinent. Figure 4.26a presents a bar chart of perceived naturalness, with the blue bins representing ground truth samples and the red bins representing synthetic/converted samples. The neutral TTS system achieved the best result, with an average score of 3.1, compared to 4.7 for the ground truth. The proposed EVC system averaged around 1.5, while the ground truth samples averaged around 4.3. Since the proposed emotional samples were converted from synthetic neutral speech, the score difference between them—approximately 1.6—merits further attention.

The fourth question in the questionnaire assesses the quality of the speech, and the results are shown in Figure 4.26b. The synthetic *Neutral* speech and the ground truth achieved scores of 4.0 and 4.6, respectively. In contrast, the proposed emotional speech attained a score of about 2.6, resulting in a difference of 1.4 compared to the neutral speech.

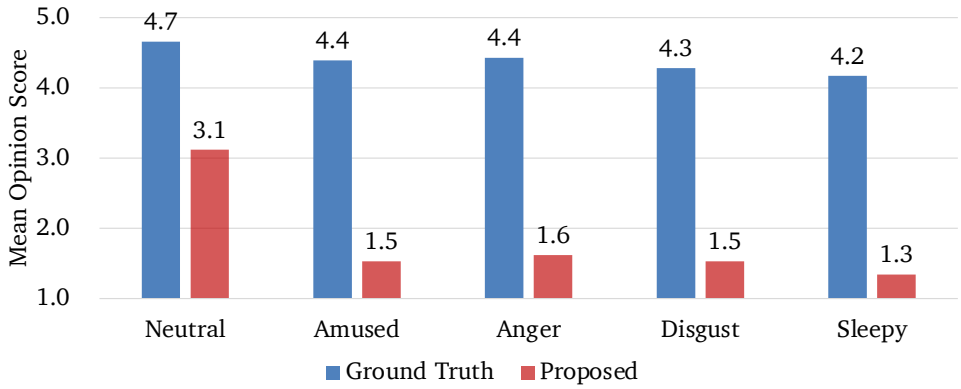
Likeability, which measures how much the participants enjoyed the speech sample, is determined by the last question in the questionnaire. As depicted in Figure 4.26c, the difference between the converted emotional and the neutral speech is 1.2, with the former scoring 1.8 and the latter 3.0. Although the likeability results are the best among the three MOS-based evaluation methods, there remains a significant gap compared to the neutral TTS system across all criteria.

In summary, the proposed EVC system has three main strengths. Firstly, despite being based on the neutral TTS system using Tacotron 2 and LPCNet, it can be directly implemented on any neutral TTS system to accomplish the ETTS task. Secondly, it offers a smaller RTF, decreasing from around 5.5 to 2.5 compared to the Dual-CycleGAN EVC system, as presented in Section 4.3.2. Lastly, the proposed EVC system is trained using a non-parallel dataset and is speaker-independent, reducing the dataset requirements compared to other parallel-trained EVC models.

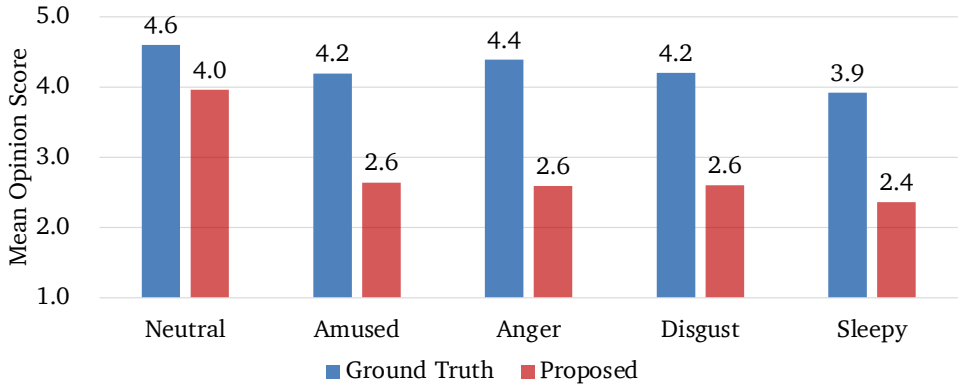
On the other hand, the system has some weaknesses. The first is that it only converts prosody and pitch at the frame level, so the converted speech and the input neutral speech have the same length. However, the pronunciation duration of the same word can vary across different emotional states. For example, *Sleepy* speech tends to be longer, while *Angry* speech tends to be shorter. This is one reason why the emotional expression does not perform well, according to the results above. The second weakness is that the proposed EVC system relies on speech-to-speech conversion, which results in a longer synthesis time (TTS + EVC) due to the two feature extraction phases and two voice synthesis phases, as explained in Section 4.4.3.1. Feature-to-feature conversion could shorten the conversion time.

Figure 4.26: Bar Charts of the MOS Test Results

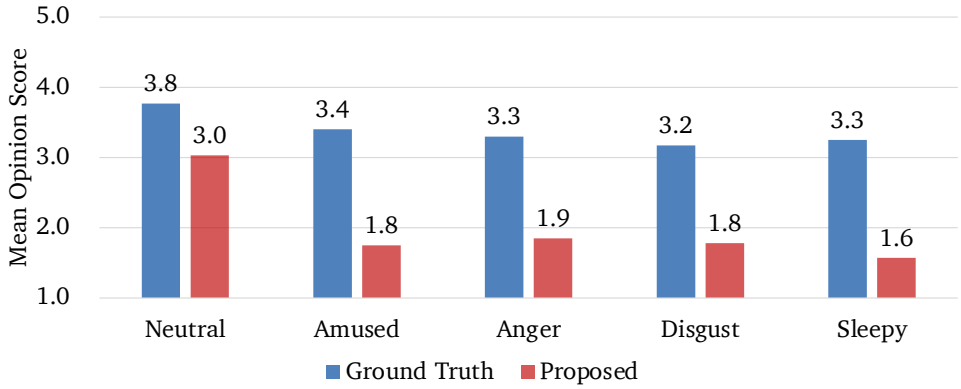
(a) Naturalness



(b) Speech Quality



(c) Likeability



Therefore, further research and improvement of the EVC technique are necessary, based on the invaluable evaluation results and discussion in this section.

4.4.3.4 Further Investigation

Besides the evaluation questions introduced in Section 4.4.3.3, several participants left extra comments on their questionnaires. The most valuable opinion was that the speech quality influences the perception of emotional expression. Therefore, to address all problems revealed by the evaluation, the highest priority should be given to optimising the speech quality. The first factor considered was the vocoder. According to the attempt described in Section 4.3.4, both prosodic and spectral features are essential rather than the sole spectrogram for EVC. Although novel vocoders based on Mel-spectrograms have achieved great performance, WORLD is still an excellent vocoder based on F_0 and SSEs. Thus, two different feature extraction functions provided by WORLD, `dio()` and `harvest()`, were compared, and `harvest()` performed better in the extraction of F_0. The converted speech using `harvest()` eliminates the clicking sound but results in a slightly longer extraction time.

The second aspect concerns the framework of the current ETTS system. The proposed EVC system converts synthetic speech generated by the neutral TTS system, and the converted samples in the subjective evaluation were produced in this way. However, the training of the EVC model is based on natural speech recordings. There are two main differences between the training and inference phases: the speaker and the naturalness. During training, the model learns the mapping between the 'natural' speech of speakers in the 'EmoV-DB' dataset. In contrast, during inference, the model is required to convert the 'synthetic' speech of the speaker in the 'Blizzard 2011' dataset. Therefore, bringing the training data closer to the data used in the inference phase is one potential way to improve speech quality. Two different experiments were designed and implemented: the first used natural samples from the Blizzard 2011 dataset as training data, while the second used synthetic samples generated by the neutral TTS system, with the speaker being the same in both training sets. Results showed that both models performed worse than the proposed EVC system, likely because both models had to learn additional information due to the speaker differences between the source and the target samples.

5

Sequence-to-Sequence Emotional Voice Conversion

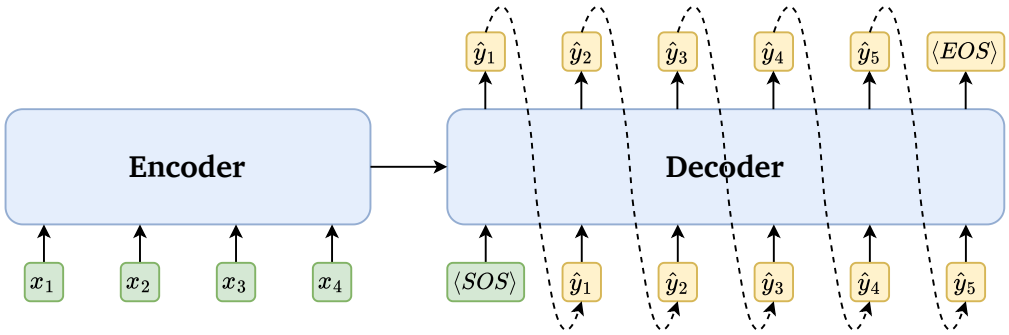
EVC is defined as the conversion of speech from one 'source' emotional state to another 'target' emotional state [78, 147]. The term 'conversion', used in both EVC and VC, has led most mainstream studies to focus on frame-to-frame methods, as introduced in Sections 4.1 and 4.2. However, the frame-to-frame method has inherent disadvantages in conversion tasks, especially in EVC, where emotional expressions are involved.

The most significant disadvantage is that emotion is inherently suprasegmental and complex, involving multiple signal attributes related to both prosody and spectrum. The frame-to-frame EVC method, particularly when converting prosody and spectrum separately, is insufficient [78]. Additionally, because of the frame-to-frame conversion, the input and output have the same number of frames, meaning that the converted speech and the source speech have the same duration. However, it is widely understood that the duration of the same sentence can vary depending on the emotional expression. In other words, speech rates differ when expressing different emotional states, even for the same speaker. Lastly, emotional expression in speech often appears only in parts or certain words, which is not always captured by the frame-to-frame method. Therefore, a new solution is necessary to address these issues currently encountered in frame-to-frame conversion, and this new solution is called sequence-to-sequence.

5.1 Sequence-to-Sequence Emotional Voice Conversion

In this section, sequence-to-sequence learning, as the core of sequence-to-sequence EVC research, is introduced first. Next, the challenges associated with applying

Figure 5.1: Architectural Framework of Sequence-to-Sequence Model



sequence-to-sequence learning are explained. Following this, state-of-the-art studies are organised and analysed from different perspectives.

5.1.1 Sequence-to-Sequence Learning

Sequence-to-sequence learning was initially proposed for machine translation tasks [39] and has since demonstrated its effectiveness in several speech processing and synthesis tasks [20, 218, 219]. A typical sequence-to-sequence model consists of two main modules: the encoder and the decoder. Unlike in a standard Autoencoder model, where the decoder generates the output at one time [176], sequence-to-sequence models generate predictions iteratively. The previous prediction, along with the encoded information, is used as the input to the decoder until the entire sequence is generated [39].

The framework diagram of a typical sequence-to-sequence model is illustrated in Figure 5.1. The source sequence $\mathbf{X} = (x_1, x_2, x_3, \cdots, x_m)$ is first processed by the encoder to extract information $\mathbf{H}$ that will be utilised by the decoder. On the other side, the decoder begins with the $\langle SOS \rangle$ token to predict the first frame $\hat{y}_1$ using the encoder's output. The predicted frame $\hat{y}_1$ is then treated as a new input for the decoder to predict the next frame $\hat{y}_2$. This inference process continues iteratively until the $\langle EOS \rangle$ token is generated, resulting in the generated sequence $\mathbf{Y} = (\hat{y}_1, \hat{y}_2, \hat{y}_3, \cdots, \hat{y}_n)$. Algorithm 2 explains the sequence-to-sequence learning process in pseudocode, clearly demonstrating that the lengths of the generated sequence and the source sequence are not necessarily equal. This flexibility allows sequence-to-sequence models to generate outputs of variable lengths, which is particularly beneficial for tasks such as machine translation between different languages [39].

Algorithm 2: Sequence-to-Sequence Learning

Input: Source Sequence: $(x_1, x_2, x_3, \cdots, x_m)$, $m \in \mathbb{R}^+$
Start of the Sequence Token: $\langle SOS \rangle$
End of the Sequence Token: $\langle EOS \rangle$
Max Length: $L \in \mathbb{R}^+$

Output: Sequence $(\hat{y}_1, \hat{y}_2, \hat{y}_3, \cdots, \hat{y}_n)$, $n \in \mathbb{R}^+$, $n \leq L$

$\mathbf{H} \leftarrow \text{Encoder}((x_1, x_2, x_3, \cdots, x_m))$
$\hat{y}_0 \leftarrow \langle SOS \rangle$
$i \leftarrow 0$
while $i \leq L$ **do**
  $i \mathrel{+}= 1$
  $\hat{y}_i \leftarrow \text{Decoder}(\hat{y}_{i-1}, \mathbf{H})$
  if $\hat{y}_i = \langle EOS \rangle$ **then**
    return $(\hat{y}_1, \hat{y}_2, \hat{y}_3, \cdots, \hat{y}_{i-1})$
  end
end
return $(\hat{y}_1, \hat{y}_2, \hat{y}_3, \cdots, \hat{y}_L)$

5.1.2 Sequence-to-Sequence in EVC

Returning to the three disadvantages of frame-to-frame EVC methods discussed earlier in this chapter, sequence-to-sequence models offer solutions to all of them, at least to some extent. Sequence-to-sequence models have demonstrated superior performance in the joint conversion of prosody and spectrum in EVC tasks [78], as compared to frame-to-frame models, which have been shown to be inadequate for this purpose in previous studies (Sections 4.3.4 and 4.4.3.1). Furthermore, sequence-to-sequence models can produce variable-length outputs, meaning the duration of the converted speech is not fixed, which can significantly enhance emotional expression.

A similar problem exists in machine translation, related to the relationship between specific words in the source sentence and the current predicted word. This issue was not fully appreciated until the attention mechanism was introduced [41], as discussed in Section 2.1.3. In the context of EVC, attention mechanisms can assign greater weight to the more relevant parts of the source speech when predicting the converted speech.

To summarise, the application of sequence-to-sequence learning in EVC offers three main advantages over frame-to-frame methods:

1. Sequence-to-sequence models can simultaneously learn feature mapping, alignment and duration prediction.

2. Sequence-to-sequence models avoid errors that arise from the separate mapping of prosody and spectrum.

3. The attention mechanism enables sequence-to-sequence models to focus on emotionally emphasised parts of the speech.

However, sequence-to-sequence training typically requires a large dataset to enable the model to learn effectively [220]. Section 3.1.3 has reviewed and analysed publicly available emotional speech datasets, revealing that few datasets contain large volumes of speech samples. Additionally, the training data must be parallel, a requirement not necessary for some frame-to-frame models [151, 156]. This is because, in sequence-to-sequence EVC, the decoder synthesises speech starting from the $\langle SOS \rangle$ token, meaning all the information the decoder needs must be provided by the encoder and the input speech. In contrast, in frame-to-frame EVC, it suffices for the model to learn the mapping of prosody and spectrum since the source speech provides some information, such as linguistic content, during the conversion.

This represents a significant challenge for the adoption of sequence-to-sequence models in EVC research, and it has so far hindered this paradigm from becoming dominant in the field. The following section will introduce and discuss six key perspectives of sequence-to-sequence EVC research, based on current state-of-the-art studies [186, 187, 221, 222, 223, 224].

5.1.3 Challenges and Attempts

The introduction of sequence-to-sequence learning to EVC research began with a study focused on the conversion of F_0 [186]. This approach employed a three-step procedure: extracting F_0, transforming it, and applying the resulting contour to the signal. This method enabled the conversion of any neutral speech to a specific emotional category. Although F_0 contains less information compared to other commonly used features, this pioneering study laid the groundwork for further research in the field.

One of the key challenges in EVC, similar to VC, is addressing mispronunciations. To tackle this, TTS was integrated into sequence-to-sequence EVC models to guide linguistic information, moving away from text supervision techniques used in VC [225] and reducing reliance on text [187]. This study also overcame the limitation of one-to-one EVC models, such as the one proposed in [186], by introducing a many-to-many EVC model. This model employed a reference speech that conveyed the target emotion, providing direction for the conversion process.

Given the limited size of available emotional speech datasets, researchers have sought to minimise the number of required training samples. For instance, manually balancing word distribution and increasing the proportion of uncommon words in a dataset have demonstrated improved training efficiency and stability, even with fewer

training samples [221]. Additionally, implementing an emotion encoder enabled the development of a one-to-many model capable of converting high-quality emotional speech.

However, manual preprocessing of datasets is both time-consuming and costly. A more efficient solution should focus on the model itself. To this end, a two-stage training strategy was proposed to enhance the performance of many-to-many EVC models using only a small-sized parallel emotional dataset [223]. In the first stage, a TTS dataset (with neutral speech samples) was used to initialise the style. This was followed by a stage of emotion training using the small-sized parallel emotional dataset.

The studies mentioned above predominantly focused on the emotional categories expressed by the converted speech, but human emotional expression also involves varying intensities. To control the intensity of emotional expression, one approach manipulated a weight that was multiplied by the emotional embedding [222]. Another approach trained the model with variations in intensity without explicit annotations, achieving this with a smaller training set than the former method [224].

5.1.4 Training Strategy

In the training phase of a classic sequence-to-sequence model, its inherent architecture introduces two significant challenges. Firstly, since the decoder generates the frame for the current timestep based on the frame generated in the previous timestep [39], errors tend to accumulate as the sequence progresses. This accumulation makes it difficult for the model to converge, especially when dealing with long sequences. Secondly, the generation process is time-consuming because each frame must wait for the previous frames to be generated before it can proceed. To address these issues, the training strategy known as Teacher-Forced Learning is commonly employed. In teacher-forced learning, instead of feeding the generated frames back into the decoder, the ground truth (target) frames are used as input. This approach prevents error accumulation by ensuring that the decoder is always working with the 'correct' input, rather than potentially flawed generated frames. Additionally, teacher-forced learning significantly speeds up the training process through parallel training since the need to wait for the sequential generation of frames is eliminated [218].

Beyond these general strategies, specific challenges in EVC necessitate specialised training approaches. One such approach is multi-task learning, which was introduced into sequence-to-sequence EVC to address issues of mispronunciation and training instability [187]. In this approach, TTS is implemented as a secondary task. The input text is encoded into a linguistic embedding by a text encoder, while the original EVC task is performed by a content encoder, which generates the linguistic embedding from the source speech. During training, the model alternates between

performing either EVC or TTS. This form of alternating multi-task learning helps the model avoid mispronunciation errors by leveraging the strengths of both tasks.

To mitigate the requirement for a large-sized training set, a two-stage training strategy was investigated for sequence-to-sequence EVC [223]. This strategy begins with style initialisation using a large neutral TTS dataset, followed by emotional fine-tuning on a smaller emotional dataset. Additionally, a pre-training task involving VC has been shown to benefit EVC due to their inherent similarities [221].

Furthermore, adversarial training has been applied to enhance the disentanglement of style/emotional information from linguistic information. This is achieved by adding an emotional classifier that helps to remove emotional information from the linguistic embedding [223]. A similar strategy involves incorporating emotion supervision training with a pre-trained SER module. Recognising that the reconstruction loss between the target and converted speech does not adequately capture human emotional perception—which is the ultimate evaluation metric in EVC—a SER module is introduced to compute two perceptual losses: emotion classification loss and emotional embedding similarity loss. These perceptual losses help optimise the emotional perception of the converted speech.

5.1.5 Model Architecture

The simplest sequence-to-sequence model architecture for EVC includes one encoder, one decoder, and one attention module. In this basic setup, the encoder processes the extracted features from the input speech and generates a context vector, which is then used by the decoder—along with the previous frames—to produce the converted features. The attention mechanism plays a crucial role by providing an explicit alignment between the input (source) and the output (converted) speech.

However, to enhance the extraction of different types of information from the speech signal, multiple encoders can be employed, with each encoder dedicated to handling a specific kind of information. For instance, a sequence-to-sequence EVC model was developed using three individual encoders: a style encoder, a content encoder and a text encoder [187]. In this architecture, the style encoder focuses on extracting style embedding from a reference speech sample, while the content encoder and text encoder are designed to extract linguistic embeddings from the source speech and the input text, respectively, facilitating multi-task learning. During TTS training, the style embedding, along with the linguistic embedding from the text, is sent to the decoder to reconstruct emotional speech. In EVC training, the source speech is used instead of the text to obtain the linguistic embedding, allowing the model to perform both TTS and EVC based on the input type after training.

The application of additional encoders also brings other advantages to the EVC model. For example, incorporating a speaker encoder, which disentangles speaker information, enables the model to utilise multi-speaker datasets during training [221,

222]. The speaker encoder extracts speaker information, which in turn enhances the extraction of other relevant information from the speech.

To further optimise the extraction of linguistic information, adversarial learning is applied by introducing an emotion classifier alongside the encoders mentioned above [223]. Although linguistic information can be obtained by the content encoder, residual emotional information may still be present in the extracted linguistic embedding, necessitating further elimination. The emotion classifier is used to ensure that sufficient speaker information is included while irrelevant emotional information is excluded.

Conversely, from a different perspective, a source decoder and a target decoder were utilised to ensure that the linguistic embedding contained the necessary information [222]. This approach helps maintain the relevance and quality of the linguistic embedding during the conversion process.

Building on this, two additional modules—an intensity encoder and a pre-trained SER module—were introduced to control emotional intensity and enhance the emotional expressiveness of the output speech [224]. Based on the assumption that emotional intensity can be viewed as the relative difference between neutral speech (zero intensity) and emotional speech, relative attributes were employed to train the emotional intensity model without explicit labels. The intensity embedding, which can be derived from reference speech or manually set, is then concatenated with the emotional embedding. The combined embedding is fed into the decoder to reconstruct the emotional speech with the desired intensity.

The pre-trained SER model also contributes to improving performance by generating two perceptual losses [224]. An emotion classification loss is computed by classifying the converted emotional speech using the SER model and comparing it with the ground truth emotional category. Additionally, an emotion embedding similarity loss is calculated by comparing the emotion embedding from the emotion encoder with the SER embedding derived from the converted speech. Visualisation of the results showed that these perceptual losses help the emotion encoder better discriminate between different emotion categories.

Moreover, other modules like the length regulator, which aligns the lengths of the encoder outputs with the decoder inputs, and the Connectionist Temporal Classification (CTC) recogniser [226], implemented after the decoder to guide alignment, have been shown to successfully improve the performance of EVC models [221]. These modules contribute to more accurate and expressive emotional speech conversion by addressing alignment and temporal aspects in the sequence-to-sequence learning process.

5.1.6 Datasets

As discussed in Section 5.1.2, a large-sized dataset is crucial for achieving good performance in sequence-to-sequence training [220]. Specifically,

sequence-to-sequence EVC training requires a large, parallel, one-speaker and emotional dataset. For example, the mKETTS dataset, a Korean emotional speech dataset, was used in EVC [187]. This dataset contains 3,000 utterances per emotional category, all spoken by one male speaker, with 7 emotional categories in total (*Neutral*, *Anger*, *Disgust*, *Fear*, *Happiness*, *Sadness* and *Surprise*).

In contrast, a smaller dataset with only 200 emotional speech samples (10 sentences × 4 emotional categories × 5 levels of intensity) was used in another study [186], which included the emotions *Anger*, *Joy*, *Fear* and *Sadness*. This small dataset was sufficient because the conversion was performed at the syllable level, with the model being trained on approximately 1,100 syllable pairs after forced alignment [227].

The inclusion of a speaker encoder has expanded the range of available datasets from single-speaker datasets to multi-speaker datasets. For instance, a Chinese emotional speech dataset featuring 3 speakers, 3 emotional categories (*Anger*, *Happiness* and *Sadness*), and a total of 6 hours of recordings was used to fine-tune a pre-trained VC model for EVC [221]. Similarly, a Korean dataset containing 100 sentences across 4 different emotional categories (*Neutral*, *Anger*, *Happiness* and *Sadness*), performed by 5 male and 5 female actors, resulting in a total of 4,000 samples, was also utilised [222].

Modifications to the training strategy have made it possible to use datasets that were originally unsuitable. For example, in the first stage of a two-stage training strategy, approximately 30 hours of recordings from 99 speakers in the VCTK multi-speaker neutral dataset [69] were used. This was followed by fine-tuning with only 350 pairs of emotional speeches from the ESD dataset [99], enabling the model to incorporate emotional expression and improve performance [223, 224].

These examples highlight the flexibility and adaptability of sequence-to-sequence EVC training, demonstrating that both dataset size and training strategies play critical roles in achieving effective emotional voice conversion.

5.1.7 Model Inputs

Most sequence-to-sequence EVC studies have utilised Mel-spectrograms as the primary acoustic feature [187, 221, 222, 223, 224], but different vocoders have been employed to synthesise the final speech output. For instance, researchers have used vocoders such as Parallel WaveGAN [228], WaveRNN [61] and HiFi-GAN [44]. An exception to this trend is found in a study focusing on syllable-level conversion, where the F_0 contour was chosen as the feature. The converted F_0 contour was then used to reconstruct the speech using the SuperVP vocoder [229].

A critical aspect of EVC is the control of the target emotion. Beyond the straightforward one-to-one EVC [186], there are three primary methods to introduce target emotion information into the model. The most direct approach involves feeding an emotion ID into the emotion encoder [221]. Alternatively,

a reference speech sample can be used during the inference phase to guide the model [187]. Another method involves using emotional embeddings, which are calculated by averaging a set of emotional embeddings from the same emotional category [222, 223, 224].

5.1.8 Evaluation Methods

Both objective and subjective evaluation methods have been employed in state-of-the-art EVC studies. Objective evaluation typically involves calculating a specific measure of difference or correlation between the output and the target. For example, WER [187, 221] and Character Error Rate (CER)[221], commonly used metrics in ASR, were utilised to assess the linguistic consistency of the emotion conversion. Additionally, other metrics focus on the acoustic perspective. For instance, MCD was used in three studies to measure the distortion between the converted and target speech[222, 223, 224]. Furthermore, VDE, GPE, FFE [222] and Difference of Duration (DDUR) [223, 224], which calculates the duration difference between the converted and target speech, were employed to evaluate the performance of the EVC model.

Subjective evaluation involves asking human participants to listen to the output samples and provide their subjective opinions based on given perspectives. For example, MOS, the most popular metric in TTS, VC and EVC, was used in five sequence-to-sequence EVC studies to assess dimensions such as clarity, naturalness and similarity [187, 221, 222, 223, 224]. Additionally, the ABX test, which asks participants to identify whether a provided sample X belongs to class A or B, was used in one study [222]. Best-Worst Scaling (BWS), another evaluation metric that identifies extreme items (best and worst) [230], was applied in a study also on clarity, naturalness and similarity [223, 224]. Moreover, subjective SER was conducted in another study, where 87 participants were asked to select the emotion category they perceived from the given speech [186].

Table 5.1: Information of Reviewed Sequence-to-Sequence EVC Papers

Paper	**Highlights**	**Feature Set & Vocoder**	**# of Samples**	**Language**	**Emotional Model**	**Evaluation Methods**
[186]	First work Syllable-level conversion	F_0 contour SuperVP	∼1 100 syllables	French	One-to-one	SSER
[187]	Multi-task learning	Mel-spectrogram Griffin-Lim	21 000	Korean	Many-to-many	WER CS MOS ABX
[221]	Redundancy reduction CTC leverage EVC fine-tuning	Mel-spectrogram HiFi-GAN	6 000	Chinese	One-to-many	WER CER MOS
[222]	Multi-speaker dataset Context preservation Emotional intensity	Mel-spectrogram Parallel WaveGAN	4 000	Korean	One-to-many	MCD VDE GPE FFE MOS ABX SSER
[223]	Two-stage training Small dataset	Mel-spectrogram WaveRNN	350	English	Many-to-many	MCD DDUR MOS BWS
[224]	Style pre-training Small dataset Emotional intensity	Mel-spectrogram Parallel WaveGAN	350	English	Many-to-Many	MCD DDUR MOS BWS

ABX: ABX test **BWS**: Best-Worst Scaling **CER**: Character Error Rate **CS**: Cosine Similarity
CTC: Connectionist Temporal Classification **DDUR**: Differences of Duration **FFE**: F_0 Frame Error
GPE: Gross Pitch Error **MCD**: Mel-cepstral Distortion **MOS**: Mean Opinion Score
SSER: Subjective Speech Emotion Recognition **VDE**: Voicing Decision Error **WER**: Word Error Rate

5.2 Application: Transformer and EmoV-DB

As described in Section 4.2.3.3, RNNs, including variants like LSTM-RNN and GRU-RNN, were designed to handle sequential data in machine learning. However, their inherent recurrent nature prevents them from processing data in parallel [231]. While CNNs are capable of computing much faster than RNNs [232], they lack the ability to effectively manage long-range dependencies in sequential data [233]. To overcome the limitations of recurrence in RNNs and convolutions in CNNs, the Transformer model architecture processes sequential data differently by relying solely on the attention mechanism [43].

5.2.1 Transformer

The architectural figure of the Transformer is presented in Figure 5.2. The most innovative aspect of the Transformer is its use of Scaled Dot-Product Attention, which is applied in the Self-Attention, Masked Self-Attention and Encoder-Decoder Attention blocks depicted in the figure. For example, given two sequences, $\mathbf{X} = (x_1, x_2, x_3, \cdots, x_m)$ and $\mathbf{X}' = (x'_1, x'_2, x'_3, \cdots, x'_n)$, their scaled dot-product attention can be computed starting with the Keys, Values and Queries:

$$\begin{aligned} k_i &= W_K \cdot x_i \\ v_i &= W_V \cdot x_i \\ q_i &= W_Q \cdot x'_i \end{aligned} \tag{5.1}$$

where W_K, W_V and W_Q are three trainable parameter matrices. Then, queries q and the key matrix $K = (k_1, k_2, k_3, \cdots, k_m)$ are used to compute the weight vector α:

$$\alpha_i = softmax(\frac{K^T \cdot q_i}{\sqrt{d_K}}) \tag{5.2}$$

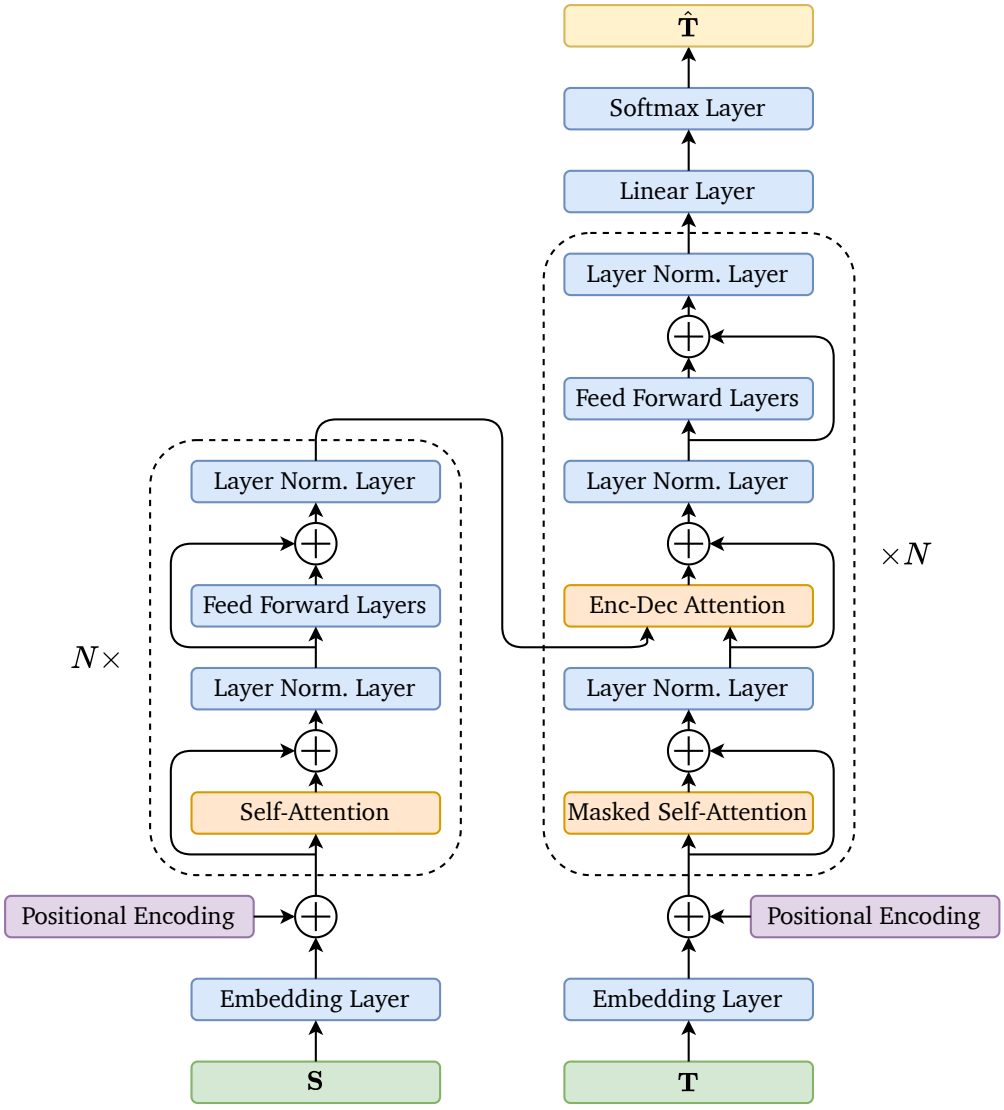
where d_K denotes the dimension of the key matrix. Using the value matrix $V = (v_1, v_2, v_3, \cdots, v_m)$ and the weight vector α, the context matrix $C = (c_1, c_2, c_3, \cdots, c_m)$, which is also the output of the attention module, is computed as follows:

$$c_i = V \cdot \alpha_i \tag{5.3}$$

To summarise, the equation for scaled dot-product attention between two sequences is expressed as:

$$attn_{sdp}(\mathbf{X}, \mathbf{X}') = softmax(\frac{(W_K \cdot \mathbf{X})^T \cdot (W_Q \cdot \mathbf{X}')}{\sqrt{d_K}}) \cdot (W_V \cdot \mathbf{X}) \tag{5.4}$$

Figure 5.2: Architectural Framework of Transformer



Alternatively, in the form of query, key, and value:

$$attn_{sdp}(Q, K, V) = softmax(\frac{K^T \cdot Q}{\sqrt{d_K}}) \cdot V \tag{5.5}$$

In the Transformer, the encoder-decoder attention is computed using the output of the encoder and the output of the masked self-attention (explained later) in the decoder. As an autoregressive generative model, the input to the decoder begins with an $\langle SOS \rangle$ token, denoted as $\hat{t}_0$. This token is then sent into the decoder, where the output is regarded as $\hat{t}_1$ and fed back into the decoder as the new input. This process continues until the $\langle EOS \rangle$ token is generated as the l-th token, or the maximum length l is reached. The final output is $\hat{\mathbf{T}} = (\hat{t}_1, \hat{t}_2, \hat{t}_3, \cdots, \hat{t}_l)$.

The purpose of applying self-attention in the Transformer is to compute a representation of a sequence by capturing the relationships and dependencies between tokens within the same sequence. The computation of self-attention is performed using one sequence instead of two; for example, in the encoder, it is computed as $attn_{sdp}(\mathbf{S}, \mathbf{S})$, where $\mathbf{S}$ is the input sequence to the encoder. However, considering the complex nature of the decoder's input, especially during teacher-forced training, information from future tokens (those to the right of the current token) cannot be used. To prevent improper usage, the values of future tokens in the attention mechanism are set to $-\infty$, resulting in no contribution from future information, and it is called masked self-attention [43].

Furthermore, to enhance the representational power of the Transformer, multiple scaled dot-product attentions are computed instead of just one, with each individual attention denoted as a Head [43]. The computation process for multi-head attention is expressed as:

$$\begin{aligned} multihead(Q, K, V) &= W_O \cdot concat(head_1, head_2, \cdots, head_h), \\ head_i &= attn_{sdp}(Q_i, K_i, V_i) \end{aligned} \tag{5.6}$$

where W_O is a parameter matrix used to project the concatenated outputs of the different heads back to the original model dimension.

Both the encoder and the decoder consist of attention layers, feed-forward layers and layer normalisation layers. Additionally, a residual architecture [234] is applied to the connections between the layers, as shown in Figure 5.2. With a linear layer and a *softmax* operation (for the original machine translation task of the Transformer) following the decoder, the main architecture of the Transformer is established.

However, without a recurrent or convolutional architecture, the model cannot leverage information from the token order in the sequence, which is essential for understanding the sequence deeply [43]. Therefore, the inputs to both the encoder and decoder need to be augmented with positional information. Instead of using the simplest discrete method $(0, 1, 2, \cdots)$ or trainable encoding, the Transformer applies sine and cosine functions to smooth variations between tokens:

$$pe(p,i) = \begin{cases} sin(\frac{p}{10000^{i/d}}), & \text{if } i \text{ is even} \\ cos(\frac{p}{10000^{i/d}}), & \text{if } i \text{ is odd} \end{cases} \tag{5.7}$$

where p and i indicate the position and the dimension, respectively. d denotes the dimension of the model. The positional encoding $pe(p,i)$ has the same dimension as the input, so they are added together and sent into the network.

Despite the excellent performance in machine learning tasks, the Transformer has several advantages:

1. **Parallelisation**. The Transformer benefits from its self-attention mechanism, allowing it to process the source sequence in parallel, which saves time compared to the sequential processing of RNNs, particularly during the training phase.

2. **Long-range Dependencies**. The Transformer can capture and learn long-range dependencies in a sequence, which is not the strength of CNNs. Even compared to RNNs, which specialise in sequential data, the Transformer achieves better performance [43].

3. **Scalability**. It is easy to scale the Transformer according to the size of the training dataset and the complexity of the task by adjusting the dimensions, the number of encoder and decoder layers, the number of heads, etc.

However, the Transformer still struggles in several aspects and under certain circumstances:

1. **Computational Complexity**. The Transformer requires substantial computational resources due to the self-attention process, especially when dealing with complex tasks.

2. **Data Efficiency**. The Transformer needs a large amount of training data to achieve convergence. While RNNs require smaller training sets, benefiting from a better understanding of sequential information.

3. **Sequential Adaptability**. Although the Transformer introduces sequential information through positional encoding, RNNs still surpass it in exploiting sequential information due to their inherent recurrence.

The Transformer has demonstrated its exceptional performance in a wide range of machine learning tasks. Besides its original application in machine translation [43, 235], other studies focusing on NLP [236, 237], CV [238, 239], CA [40, 240] and multimodality [241, 242] have achieved remarkable results using the Transformer.

Moreover, Large Language Models (LLMs)—the most powerful application of neural networks—are also built on the Transformer architecture [243, 244]. Therefore, in this section, the Transformer is selected as the fundamental architecture to explore the capability of the sequence-to-sequence model in the EVC task.

5.2.2 Pilot Experiment

This section begins the exploration of Transformer-based models within the EVC field. A pilot experiment was designed to evaluate the performance of the Transformer on a parallel emotional speech dataset, focusing specifically on a one-speaker conversion from *Neutral* to *Angry* emotion.

Given the demonstrated success of Mel-spectrograms in state-of-the-art systems [187, 221, 222, 223, 224], the 80-dimensional Mel-spectrogram was selected as the acoustic feature. The high complexity of the Transformer, while beneficial for capturing complex patterns, also demands significant computational resources. This complexity justified the choice to replace the conventional 'F_0 + MCEPs/SSEs' scheme with the Mel-spectrogram in this experiment.

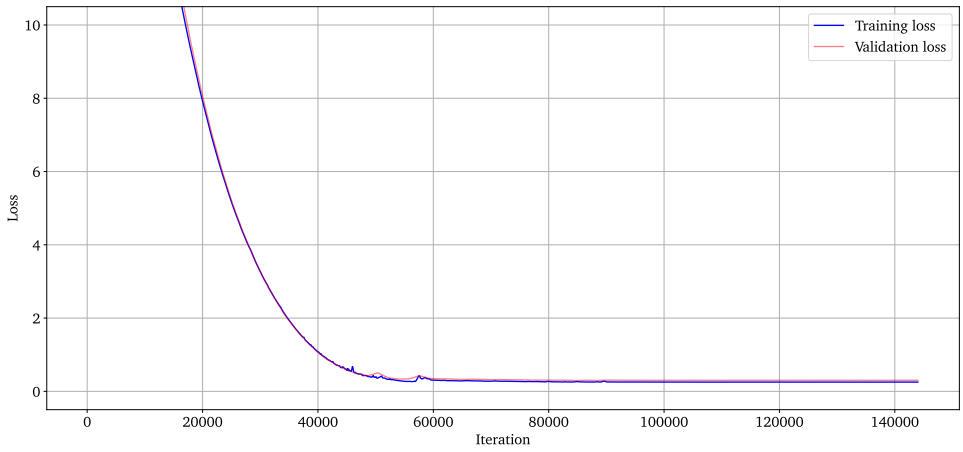
For the model architecture, the original Transformer configuration [43] was largely retained, including 6 encoder layers, 6 decoder layers, and 8 attention heads, to mitigate the influence of model complexity on the performance. However, the dimension of the source sequence was adjusted to 80, in contrast to the original 512 dimensions. The MSE loss function was employed, and a teacher-forced training strategy was used to accelerate the training process.

Data for this experiment was sourced from the EmoV-DB dataset [100], specifically focusing on the *Neutral* and *Angry* samples of the speaker *Spk-Je*. Given the noted issues with speech quality (as discussed in Section 3.2.4), a strict data cleaning procedure was conducted. This process addressed problems such as incorrect ID-sample pairings and inaccurate transcripts. After manual selection, *Spk-Je* was chosen over another female speaker due to a more balanced distribution of *Neutral* and *Angry* samples. The initial count of 417 *Neutral* and 496 *Angry* samples was reduced to 364 pairs post-cleaning. The dataset was then split into a training set (287 pairs), a validation set (72 pairs) and a test set (5 pairs).

The Adam optimiser was employed with a learning rate of 1×10^{-4} and a decay rate of 0.1. The training was conducted on a single NVIDIA® TITAN X graphics card, with a batch size of 8, reflecting the computational demands of the Transformer. The results of this pilot experiment are presented in Figure 5.3.

During training, both the training and validation losses decreased significantly within the first $40,000$ iterations. The lowest validation loss recorded was 0.295 MSE after $84,600$ training iterations, with the training loss at 0.259. However, despite the low MSE, the perceptual quality of the reconstructed speech (by using HiFi-GAN [44]) was poor, with the converted speech lacking linguistic clarity. This suggests that while the Transformer model effectively learned the acoustic features,

Figure 5.3: Loss Curves of Pilot Experiment of Transformer EVC with EmoV-DB



the reliance on teacher-forced learning may have led to overfitting, as evidenced by the model's failure to generalise well without teacher forcing, where the validation loss plateaued around 30.

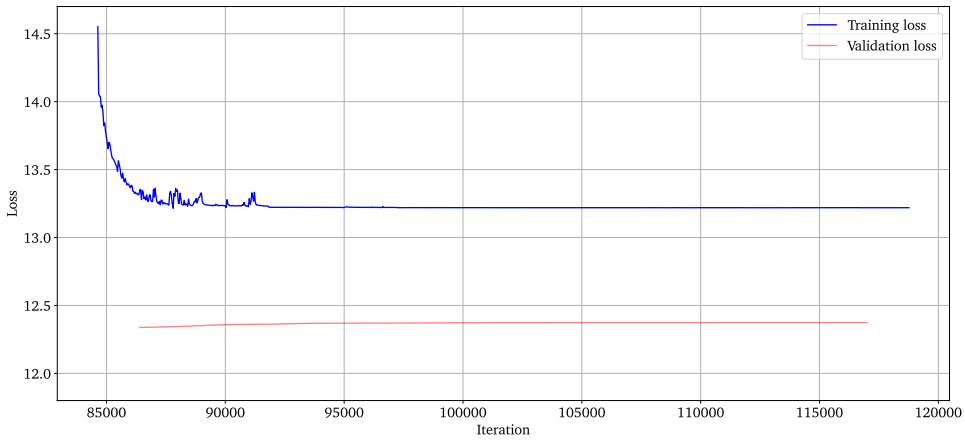
5.2.3 Optimisation of Training Strategy

Despite the generalisation drawbacks of teacher-forced learning, it offers the advantage of accelerating the training process in Transformer models by enabling parallel computation. Additionally, by using ground truth data instead of generated outputs, this approach helps prevent the accumulation of errors during the step-by-step generation process, leading to more efficient parameter updates [245]. To address the issue of generalisation while maintaining the error-prevention benefits, several strategies were implemented and tested. These strategies aimed to enhance the generalisation capability of the model without sacrificing the advantages provided by teacher-forced learning.

5.2.3.1 Non-Teacher-Forced Fine-Tuning

The first approach to improving performance involved fine-tuning the pre-trained model, initially trained with teacher-forced learning, using non-teacher-forced learning. Essentially, this process removes the dependency on ground truth during fine-tuning. Given that the pilot experiment showed the model had successfully learned to convert the ground truth into the target, the next step was to train the model to perform the task during the inference phase, using the converted Mel-spectrogram to generate the subsequent frame. The best-performing model from the pilot experiment, trained for $84,600$ iterations ($2,350$ epochs), was selected

Figure 5.4: Loss Curves of Non-Teacher-Forced Fine-Tuning of Transformer EVC



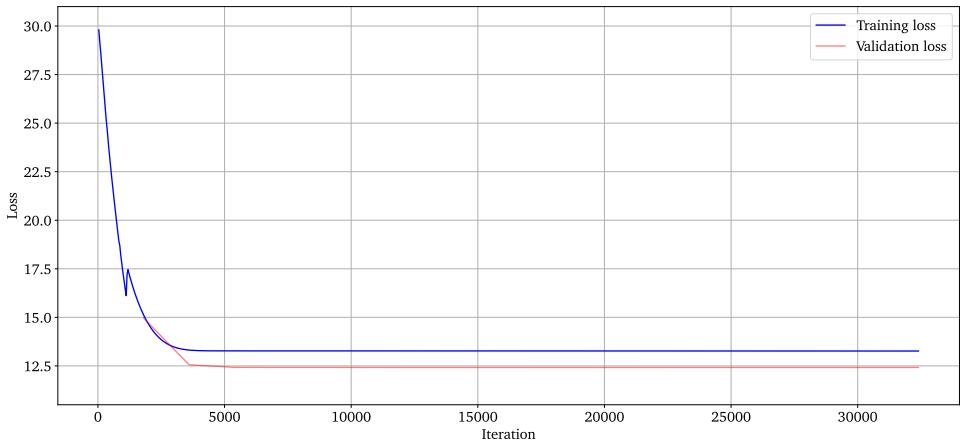
for fine-tuning without teacher-forced learning. The results of this fine-tuning are presented in Figure 5.4.

As depicted in the figure, the absence of ground truth data initially caused the training loss to increase from 0.259 to 14.551, after which it gradually decreased, as anticipated. However, the minimum loss value only converged to 13.215, which is significantly higher than that in the pilot experiment. A similar pattern was observed in the validation loss, which initially dropped to 12.338 but then showed an upward trend. This outcome indicates that the pre-trained model did not benefit from fine-tuning with generated features in an autoregressive manner. It also confirms the earlier hypothesis that the pre-trained model was overly reliant on ground truth data.

5.2.3.2 Semi-Teacher-Forced Learning

The key to successfully transitioning the model from training to inference lies in addressing the differences between the ground truth frames and the generated frames. To reduce the model's reliance on ground truth, Semi-Teacher-Forced Learning was introduced, a technique previously applied in the study of autoregressive End-to-End ETTS models [97]. Semi-teacher-forced learning incorporates both the ground truth and the generated frames to produce the next frame. Mathematically, it sends the mean of the ground truth and generated frames into the decoder, rather than relying solely on the ground truth. The equations describing teacher-forced learning, semi-teacher-forced learning and the inference phase are expressed as follows:

Figure 5.5: Loss Curves of Semi-Teacher-Forced Training of Transformer EVC



$$\begin{aligned} \text{Teacher-Forced:} \quad & \hat{y}_{n+1} = model(y_n) \\ \text{Semi-Teacher-Forced:} \quad & \hat{y}_{n+1} = model(\frac{y_n + \hat{y}_n}{2}) \\ \text{Inference:} \quad & \hat{y}_{n+1} = model(\hat{y}_n) \end{aligned} \tag{5.8}$$

where y_n and $\hat{y}_n$ represent the ground truth frame and the generated frame at time step n, respectively.

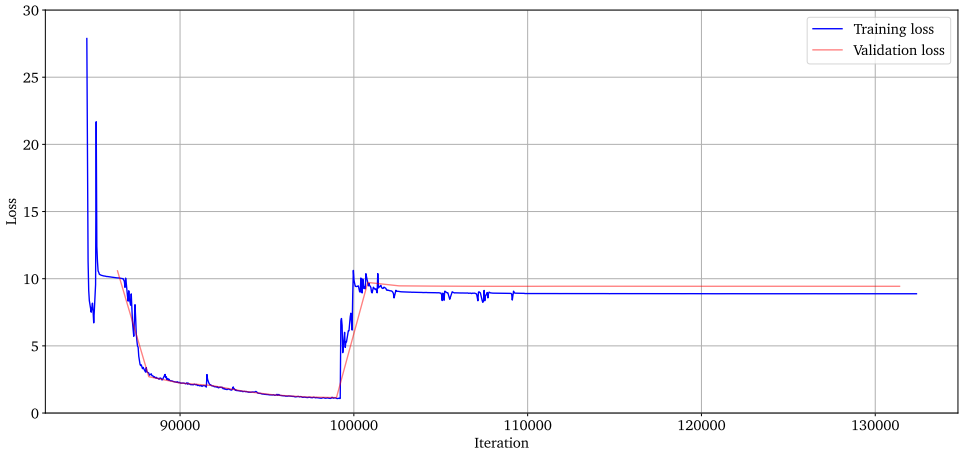
An experiment was conducted with the same configuration as the pilot experiment, except for the training strategy, and the results are shown in Figure 5.5. Since the model was trained from scratch, the training loss decreased significantly from 29.799 to a minimum of 13.266, while the validation loss converged from 14.966 to 12.420. However, this MSE value indicates that the model did not generalise well to the validation set despite using semi-teacher-forced learning. This outcome suggests that while semi-teacher-forced learning helps mitigate over-reliance on ground truth and improves generalisation to some extent, it still falls short in terms of fitting the validation data, particularly with a small training set.

5.2.3.3 Semi-Teacher-Forced Fine-Tuning

The final training strategy employed was fine-tuning the pre-trained model using semi-teacher-forced learning. The same pre-trained model described in Section 5.2.3.1 was utilised, and the results are illustrated in Figure 5.6.

After experiencing severe oscillation at the start, both the training and validation loss curves eventually declined to relatively low values, although slight rising trends were observed thereafter. The fine-tuned model, which exhibited the lowest validation loss, was achieved after an additional $14,400$ iterations $(99,000 - 84,600)$,

Figure 5.6: Loss Curves of Semi-Teacher-Forced Fine-Tuning of Transformer EVC



reaching a loss value of 1.147. This outcome demonstrates a significant improvement compared to the non-teacher-forced fine-tuning strategy. However, the results remain unsatisfactory in terms of speech reconstruction from a human perceptual evaluation. Additionally, the training speed was notably impacted by the use of generated frames, as the generation of each frame required waiting for the computation of all preceding frames.

While the aforementioned experiments on training strategies have yielded some results, the overall performance in terms of human perception remains unproductive. Although the models perform well with ground truth during training, their performance deteriorates drastically when ground truth is removed. This indicates the necessity for alternative strategies that specifically address the gap between the training and inference phases.

5.2.4 Optimisation of Data Augmentation

The Transformer model has demonstrated its powerful capability in converting ground truth to the target, as evidenced by the experimental results discussed earlier. However, the model's heavy reliance on ground truth during training highlights a significant limitation. Specifically, the model's performance is highly dependent on the size of the dataset, with larger datasets offering greater potential for optimisation. Given the strict limitation on the availability of parallel emotional speech datasets, employing data augmentation techniques becomes a more practical approach in this context.

Data augmentation encompasses a variety of techniques that artificially increase the size of a dataset based on existing data, thereby providing more training samples for neural networks [246]. This approach has proven effective in mitigating the overfitting problem that often arises from insufficient training data [247]. In the

field of speech processing, several commonly used data augmentation methods exist, including adding noise, time masking, time warping, time stretching, time shifting and filtering [248]. Additionally, generative neural networks offer another approach to synthesise samples for augmenting training data [162].

It is important to note, however, that while data augmentation methods can effectively enlarge a dataset, each method may also introduce negative effects on model training. For example, time shifting has been successful in addressing issues and performing well in recognition tasks [249]. However, in generative tasks—particularly those that require a parallel dataset—time shifting can create more challenges than it resolves. Therefore, careful consideration must be given to the choice of data augmentation methods. In this study, several data augmentation techniques were explored, with adding noise being the first method investigated.

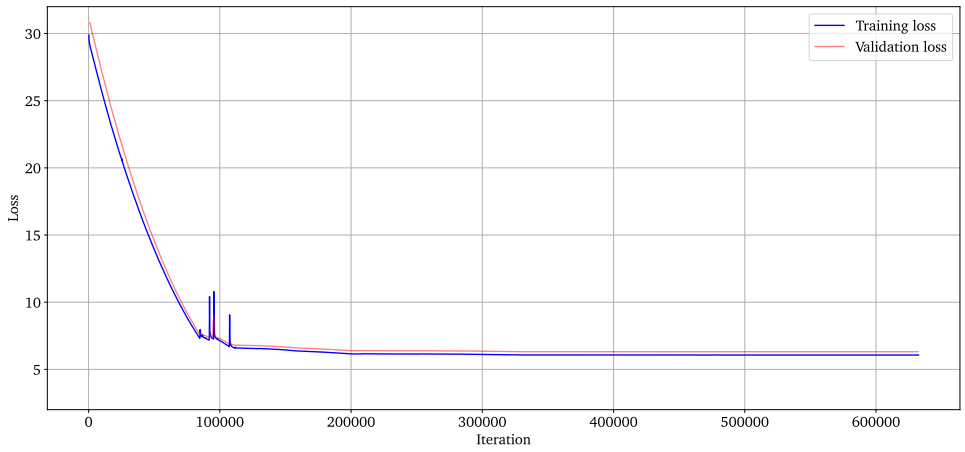
5.2.4.1 Adding Noise

Adding noise as a data augmentation technique has been well-explored in ASR research, where it has been shown to enhance performance, particularly in terms of reducing the WER [250]. While noise can negatively impact the quality of converted speech, its influence can be readily assessed and controlled through human perception. To increase the dataset size, Gaussian noise with four different SNRs was applied to all samples from the EmoV-DB dataset. This augmentation expanded the dataset by five times. The selected SNRs, ranging from 46 to 52 with an interval of 2, were chosen to ensure that the augmented samples remained clear and easily intelligible, creating a noticeable difference from the original samples.

Given the promising results of semi-teacher-forced learning from previous experiments, this method was used to train the model from scratch. Although semi-teacher-forced learning slows down the training process compared to teacher-forced learning, the primary goal here is to generate intelligible speech. The training results are presented in Figure 5.7.

The data augmentation approach using noise significantly improved model performance compared to the experiment without augmentation (Figure 5.5). The training loss dropped to a minimum of 6.068, and the validation loss reached 6.312 MSE after approximately $430,000$ iterations. In contrast, the baseline model without data augmentation only managed to achieve a validation loss of 12.420. This experiment demonstrates that data augmentation can effectively optimise the model, although further investigation into additional augmentation methods is necessary. Despite the improvements, the converted speech with an MSE of 6.312 remains unintelligible, indicating that noise alone is insufficient for producing high-quality speech and that other techniques should be explored.

Figure 5.7: Loss Curves of Transformer EVC with Adding Noise



5.2.4.2 Time Masking and Removal

The second data augmentation method implemented in this study was time masking, a technique introduced to ASR research through methods like SpecAugment [251]. In SpecAugment, time masking is performed consecutively by selecting a starting point and a duration, both sampled from a uniform distribution within a preset mask parameter. Time masking has been shown to be more efficient than other methods like time warping and frequency masking in ablation studies.

However, directly applying this consecutive time masking to a generative task like EVC presents challenges. The primary concern is that consecutive masking disrupts the parallel alignment between source and target speech samples, a critical aspect of the conversion process. Additionally, as an autoregressive model, the Transformer relies on accurate prediction of $\langle EOS \rangle$ token to cease generation, and the applied masking could interfere with this prediction.

To address these concerns, an alternative scheme was adopted: removing 5 % of frames from the Mel-spectrogram instead of masking them. These frames were selected randomly, and this process was repeated four times to create an augmented dataset equal in size to the previous one. The discrete frame removal method is better suited to the generative nature of the task, and the processed speech remains intelligible and of decent quality.

The results of this experiment, depicted in Figure 5.8, indicate that time removal outperforms noise addition. The training loss reached a minimum of 4.635, compared to 6.068 achieved by adding noise. Similarly, the validation loss for time removal was lower, converging to 4.460 after approximately $950,000$ iterations.

In addition to time removal on the Mel-spectrogram, a similar approach was tested, involving time removal on the waveform itself. However, human perception tests revealed that removing 5 % of the waveform had a more negative impact on

Figure 5.8: Loss Curves of Transformer EVC with Removing Mel-Spec. Frames

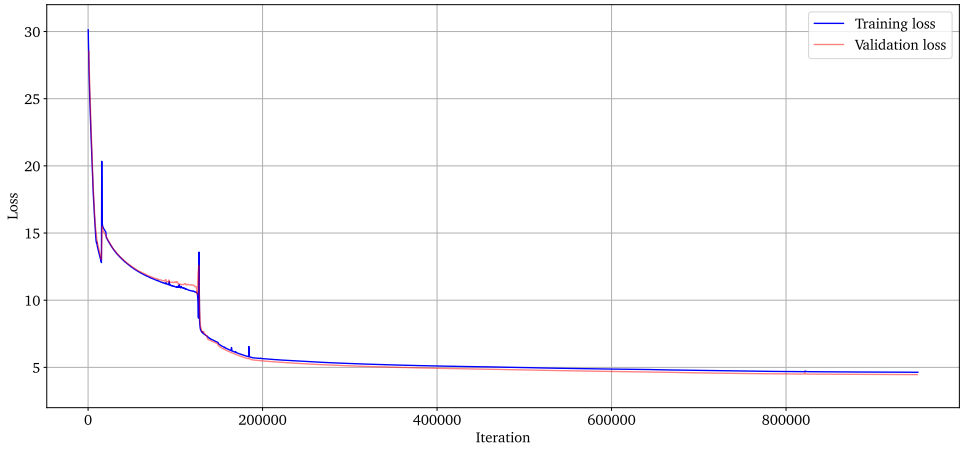
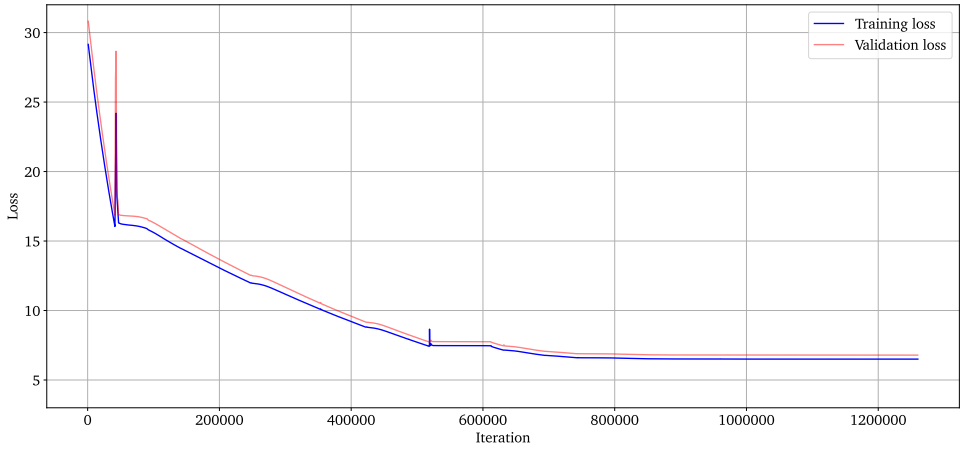


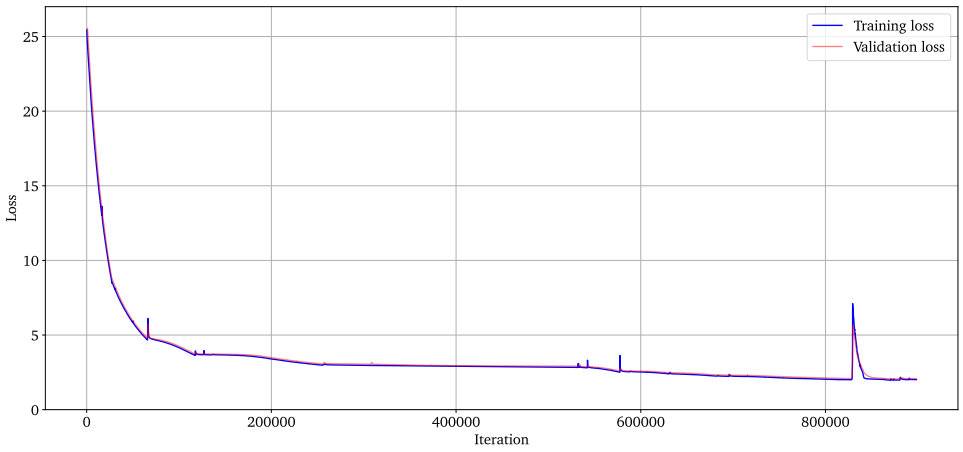
Figure 5.9: Loss Curves of Transformer EVC with Removing Waveform Frames



speech quality than removing frames from the Mel-spectrogram. To mitigate this, 1 % of the waveform was randomly removed, and this process was repeated four times, resulting in an augmented dataset five times the size of the cleaned EmoV-DB.

The results for time removal on the waveform, presented in Figure 5.9, show that this approach performed worse than noise addition. The training loss converged to an MSE of 6.502, while the validation loss reached a minimum of 6.790. The poorer performance can be attributed to the additional processing required to extract the Mel-spectrogram from the modified waveform, which introduces more uncontrolled modifications to the acoustic features used for model training.

Figure 5.10: Loss Curves of Transformer EVC with Time Stretching



5.2.4.3 Time Stretching

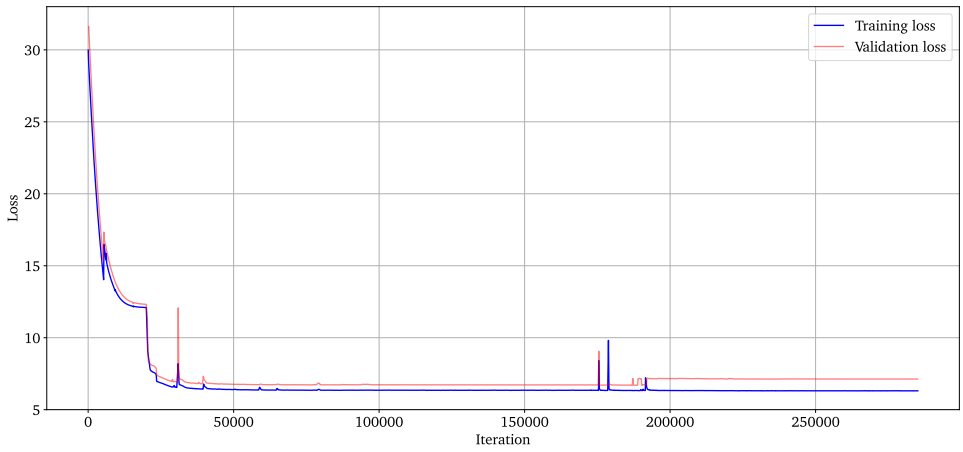
In the context of emotional speech, the speech rate is a critical factor for conveying emotional expression, making temporal data augmentation methods such as time warping and time stretching particularly cautious to implement. While methods like SpecAugment use time warping on selected parts of the Mel-spectrogram, this approach is not ideal for generative tasks like EVC. The challenge lies in finding a method that minimally affects emotional expression while producing sufficiently diverse new samples.

To address this, a linear time stretching technique was applied to the entire speech sample with ratios of $[0.90, 0.95, 1.00, 1.05, 1.10]$, resulting in an augmented dataset comparable in size to those generated by the other methods previously discussed. The results of this time-stretching approach are shown in Figure 5.10.

Among the data augmentation techniques tested, time stretching yielded the best performance. The training loss started at 25.435 and steadily decreased to a minimum of 1.972. The validation loss also reached an impressive low of 2.058 after approximately $880,000$ iterations.

However, despite these promising quantitative results, a human perception test revealed that the intelligibility of the converted speech was still insufficient. This suggests that while time stretching effectively enhanced the model's performance according to the loss metrics, it did not fully address the challenges of generating clear and intelligible speech, highlighting the ongoing need for more refined data augmentation methods or alternative strategies in the training process.

Figure 5.11: Loss Curves of Lighter Transformer EVC



5.2.5 Optimisation of Model Architecture

In neural networks, increasing model depth can often lead to overfitting, particularly when dealing with smaller datasets, which can subsequently degrade performance. This has prompted the exploration of reducing model complexity as a potential solution, especially for training with limited data. In the experiments discussed earlier, the default Transformer configuration—comprising 6 encoder layers, 6 decoder layers and 8 attention heads—was applied uniformly. However, given the constraints of the EmoV-DB dataset, even with data augmentation, such a deep model may have hindered performance.

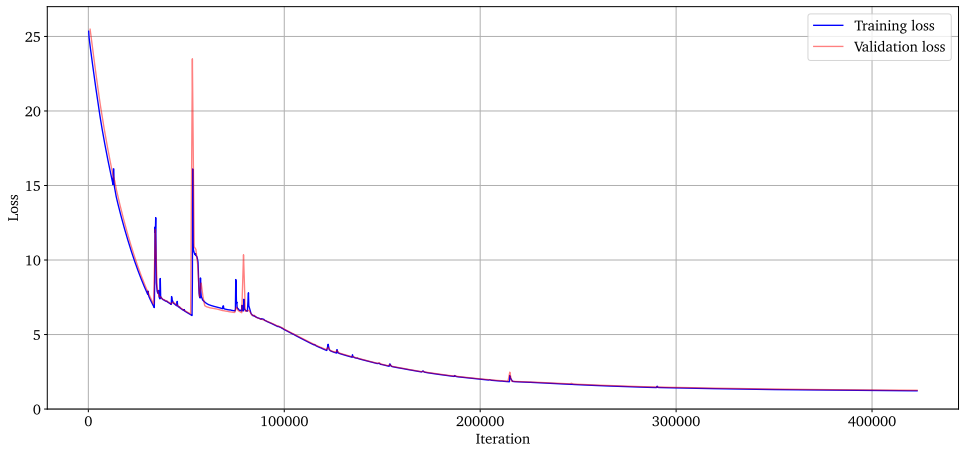
To address this, the model complexity was reduced by decreasing both the encoder and decoder layer numbers to 4 and the number of attention heads to 4. Other configurations remained consistent with the previous experiments to facilitate a clear comparison. The results of this experiment are depicted in Figure 5.11.

In the initial experiment with the default Transformer configuration, the best validation performance achieved was an MSE of 12.420 using semi-teacher-forced learning. Conversely, the lighter model improved this significantly, reducing the validation MSE to 6.695 after approximately $944,000$ iterations. This suggests that a lighter model configuration is indeed more effective when working with a small dataset, offering better generalisation and training efficiency.

Following this promising result, the lighter model configuration was applied to the augmented dataset generated by time stretching. This experiment, using the same settings as described earlier in the context of data augmentation, yielded the results shown in Figure 5.12.

As anticipated, the Transformer benefited from both data augmentation and the lighter architecture, resulting in the best performance observed across all experiments implementing semi-teacher-forced learning. After $423,000$ iterations

Figure 5.12: Loss Curves of Lighter Transformer EVC with Time Stretching



(or 2,350 epochs), the training loss reached a low of 1.224, with the validation MSE at 1.261. This represents the most effective result achieved, surpassing previous efforts with both the original and augmented datasets.

To further evaluate whether a lighter model architecture could improve performance across various data augmentation methods, the same lighter Transformer was tested with the augmented datasets previously discussed. The results, summarised in Table 5.2, reveal that the lighter Transformer improved performance primarily with the original and time-stretched datasets. Conversely, the methods involving time removal did not yield better results; in fact, the MSE increased when the model size was reduced, particularly with time removal on the Mel-spectrogram. This can be attributed to the disruption of speech signal continuity, which poses a challenge for a lighter generative neural network in predicting future frames. This trend underscores the importance of preserving continuity in generative tasks, as evidenced by the superior performance of the time-stretched dataset with the lighter model, similar to the original dataset.

Despite these advances, the intelligibility of the converted speech remains a challenge. While the model achieved low MSE on the validation set, this success was partly due to the semi-teacher-forced learning strategy. The significant drop in performance when ground truth is removed from the decoder input highlights the need for strategies that can bridge this gap. By mitigating or eliminating the reliance on ground truth during inference, the model's performance could improve, potentially leading to more intelligible speech outputs.

5.2.6 Optimisation of Scheduled Sampling

The phenomenon observed in the previous experiments is not an isolated case but is known as Exposure Bias. This term describes a situation where an autoregressive

Table 5.2: Results of the Lighter Transformer with Data Augmentation Methods

	Model	Original	AN	TR-M	TR-W	TS
Training	*Default*	13.266	6.068	4.635	6.502	1.972
	Lighter	6.303	—	6.236	6.798	**1.224**
Validation	*Default*	12.420	6.312	4.460	6.790	2.058
	Lighter	6.695	—	6.134	6.674	**1.261**

AN: Adding Noise
TR-M: Time Removal on Mel-spectrogram
TR-W: Time Removal on Waveform
TS: Time Stretching

model is trained exclusively using the ground truth from the training set, rather than its predictions, which is common in teacher-forced learning. The primary consequence of exposure bias is that errors tend to accumulate during generation when the model relies on its own predictions, leading to degraded performance. Although semi-teacher-forced learning attempts to address this by incorporating model predictions during training, the reliance on ground truth has still been evident in the experiments. Therefore, techniques that address exposure bias must be explored and implemented.

One such technique is Scheduled Sampling, which has been shown to mitigate the effects of exposure bias and improve the performance of Transformers. Scheduled sampling, similar to semi-teacher-forced learning, encourages the model to use its predictions during training. However, unlike semi-teacher-forced learning, which averages the ground truth and predictions, scheduled sampling operates at the frame level, requiring two decoders that share parameters.

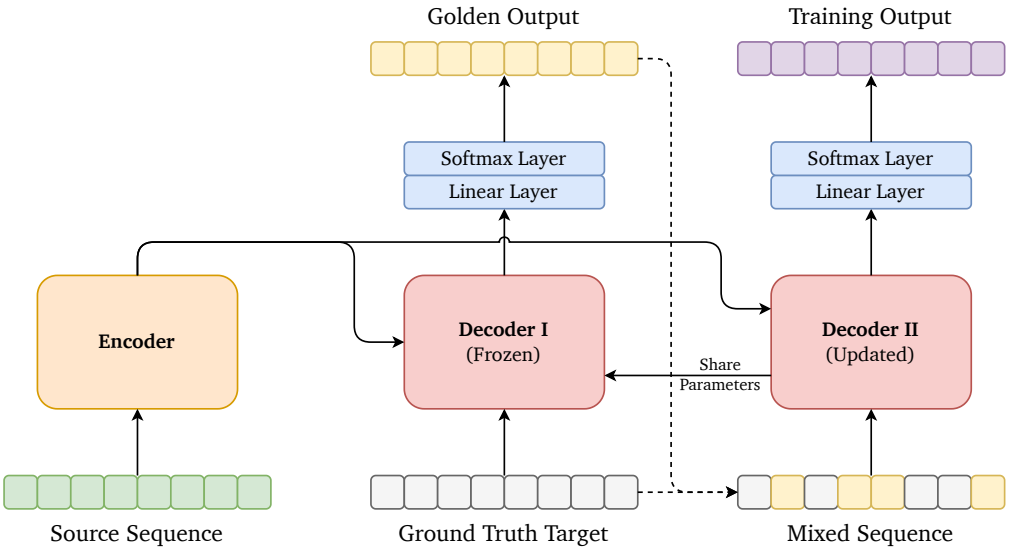
The scheduled sampling process involves the following steps:

1. **Encoding**. Similar to the standard Transformer process, the encoder generates an output from the source sequence, which is then fed into both decoders.

2. **Ground Truth Prediction**. The first decoder predicts the Golden Output using the encoder output and the ground truth target. However, this decoder is not updated at this stage.

3. **Decoder Input Mixing**. The golden output and ground truth target are mixed at the frame level according to a predefined mixing ratio, which determines the proportion of ground truth frames.

Figure 5.13: Illustration of the Application of Scheduled Sampling in Transformer



4. **Mixture Training**. The second decoder, which is updated during training, uses this mixed sequence alongside the encoder output to make predictions. The parameters of the second decoder are shared with the first.

Scheduled sampling offers a distinct advantage over semi-teacher-forced learning in terms of efficiency. While semi-teacher-forced learning requires predictions for each frame sequentially, scheduled sampling only needs two predictions per batch—one from the first decoder for the golden output, and one from the second decoder for the final output.

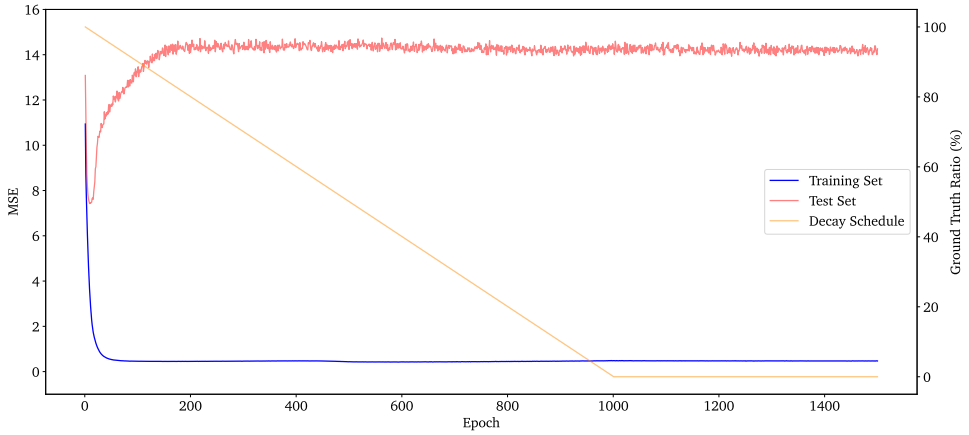
A crucial aspect of scheduled sampling is the Decay Schedule [252], which smoothly transitions the model from relying on ground truth during training to using its predictions. The linear decay schedule is one such approach, where the proportion of ground truth frames decreases gradually, eventually reaching zero. This method aims to eliminate the dependency on ground truth as training progresses.

In addition to scheduled sampling, Voice Activity Detection [1] was applied to further clean the dataset, leaving 361 pairs of *Neutral* and *Angry* speech samples. A subset of 355 pairs was used for training, with the remaining 6 pairs reserved for the test set. To maintain consistency between the training and inference phases, as well as aim for more intuitive validation results, the autoregressive method was employed during validation.

Given the success of time stretching as a data augmentation technique, an extended version of the dataset was created by stretching speech samples with ratios

[1] `https://pytorch.org/hub/snakers4_silero-vad_vad/`

Figure 5.14: Loss Curves of Transformer EVC with Linear Decay Schedule



ranging from 0.90 to 1.10, in intervals of 0.02. This resulted in a significantly larger training set of 3,905 pairs.

The final modification was made to the model architecture. Due to the similarity between the VC and EVC tasks, a Transformer-based VC model was reproduced and implemented for the current task to optimise performance [219]. Compared to the previous Transformer, the current model applies a prenet to both the encoder and the decoder, where the prenet consists of two linear layers and increases the input dimension from 80 to 256. Correspondingly, a postnet is utilised to transform the decoder output back to the 80-dimensional Mel-spectrogram. To assist in the training of the encoder and decoder, a source decoder and a target decoder are respectively connected to their outputs, attempting to restore the source input and the target from the output features. Finally, guided attention loss is introduced to the model for better alignment between the encoder and the decoder.

Although there are many modifications to the model, the key focus is the effectiveness of the scheduled sampling. The first implemented decay schedule is the Linear Decay Schedule [252]. Let ϵ be the proportion of the ground truth frames:

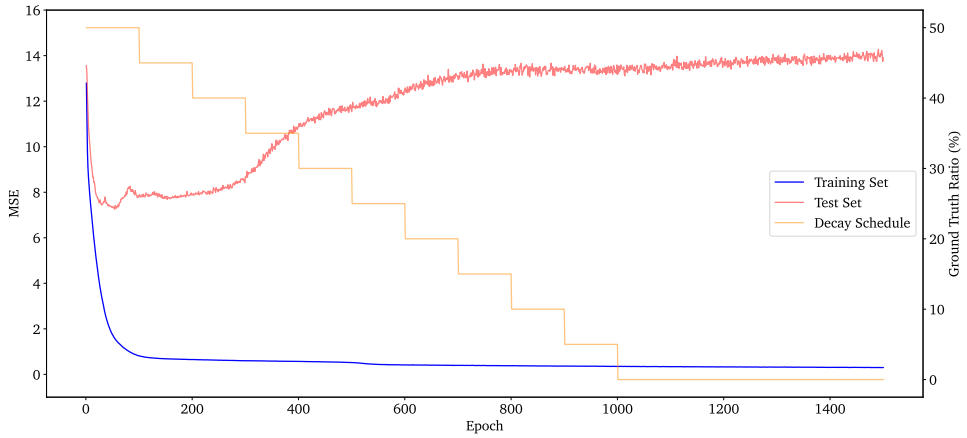
$$\epsilon = \frac{\text{number of ground truth frames}}{\text{number of all frames}} \tag{5.9}$$

$$\text{Linear Schedule: } \epsilon = max(0,\, 1 - 0.001t) \times 100\% \tag{5.10}$$

where t denotes the t-th epoch in the training phase. According to Equation 5.10, the proportion of the ground truth will linearly decrease from 100 % to 0 over 1,000 epochs. The result is shown in Figure 5.14, where the orange line indicates the decay schedule in the ground truth ratio.

The results show that the discrepancy between training and inference (test) is significant, even though scheduled sampling was applied. The MSE of the training

Figure 5.15: Loss Curves of Transformer EVC with Stepped Decay Schedule



set decreases dramatically and reaches a very low level after only a few epochs. This performance was expected since the model was basically trained using teacher-forced learning with around 100 % of the ground truth ratio. However, the MSE of the converted test speech reaches 7.420 at the 9th epoch, then increases and eventually stabilises around 14. This is a typical manifestation of overfitting, where the model relies excessively on the ground truth.

Therefore, another experiment was designed and implemented, applying a Stepped Decay Schedule:

$$\text{Stepped Schedule: } \epsilon = max(0,\, 0.5 - 0.05(\lfloor t/100 \rfloor)) \times 100\% \tag{5.11}$$

where the operation $\lfloor a/b \rfloor$ indicates integer division. Under this schedule, the ground truth ratio starts at 50 %, then decreases by 5 % every 100 epochs, eventually reaching 0 after 1,000 epochs. The stepped schedule aims to train the model at the same ratio for a longer period. The starting ratio was set to 50 % to alleviate the model's dependency on the ground truth. The result of the stepped decay schedule is shown in Figure 5.15.

As anticipated, the MSE curve of the training set is flatter than in the previous experiment, as the participation of the ground truth in model training is reduced. Additionally, the MSE curve of the test set also changes: the MSE drops during the first 50 epochs, remains roughly flat for about 250 epochs, and then increases as the ground truth ratio decreases. This result indicates that the stepped decay schedule's objective has been achieved.

However, the lowest MSE of the test set does not improve significantly compared to the linear decay schedule, where the MSE decreases from 7.317 to 7.118. This result is not significant enough, and an MSE of 7.118 on the Mel-spectrogram is not sufficient to reconstruct intelligible speech. The experimental results in this section demonstrate that simply modifying the training strategy cannot resolve model

overfitting. Improving training with limited resources still requires modification of the training data.

5.3 Application: Information Disentanglement

Currently, the overfitting problem is primarily caused by the limited size of the training set, which originally includes fewer than 300 pairs of samples. These sample pairs are drawn from one speaker and one emotional pair (*Neutral*/*Angry*). However, the dataset contains many more samples that are not being utilised. For instance, considering the newer ESD dataset, which includes 10 speakers, 5 emotional categories and 350 samples, this results in $10 \times 5 \times 350 = 17,500$ qualified samples (excluding the Chinese samples). If all these samples can be utilised in training, the model would likely benefit significantly from the expansion of the training data.

5.3.1 Information Disentanglement

In Section 5.2, the model aimed to convert a single speaker's speech from *Neutral* to *Angry*. Since the Mel-spectrogram was chosen as the acoustic feature, it is a conventional approach to convert the Mel-spectrograms of the speaker to achieve this goal. If emotional expression can be converted independently, speech samples from other speakers can also be used to train the emotional conversion model, as the speaker information is neither involved nor altered. For the same reason, the transcripts of the source and target speech do not necessarily need to match. In other words, the constraint of parallel sample pairs is removed in sequence-to-sequence EVC.

Moreover, converting the Mel-spectrogram solely to alter emotional expression can be seen as an overly complex approach. Since the Mel-spectrogram is the only feature used, and it can be reconstructed into speech by a vocoder, it contains nearly all the information of the speech. However, in this task, only the emotional expression needs to be processed. Therefore, the model can shift from the complex task of modelling the entire Mel-spectrogram to focusing on the simpler task of modelling emotional expression in speech.

To summarise, a potential strategy to utilise more training samples in the dataset is to separate the emotional information from the source speech, convert it to another emotional category, and reconstruct the output speech using the converted emotional information along with the other information from the source speech. Although different studies use varying terminologies for this separation process [221, 222, 223, 224, 253], in this section, this process is referred to as Information Disentanglement.

The application of information disentanglement is based on the assumption that speech characteristics comprise several types of information, including

speaker information, style information (e.g., speech rate, prosody) and emotional information [222]. Therefore, emotional information can be extracted from speech and converted independently. For example, emotional information can be extracted and averaged across the same category as an emotional vector. The target emotional vector, speaker vector (encoded by speaker ID) and the extracted source linguistic information are then used to reconstruct the converted speech [222]. This 'speaker + linguistic + emotional' scheme is also employed in other state-of-the-art studies [221, 253]. Additionally, a simpler binary 'linguistic-emotional' scheme is used in one-speaker EVC studies that also achieve excellent results [223], even in controlling emotional intensity [224].

Given the success of applying information disentanglement in sequence-to-sequence EVC, the 'speaker + linguistic + emotional' scheme is selected as the system architecture. Moreover, all three representations are expected to be extracted from the speech, rather than relying on ID-based representation [221, 222]. To accomplish this, three encoders, each dedicated to extracting one specific representation, are required instead of a single encoder.

Unlike previous experiments where the emotional state is converted to another state directly, information disentanglement enables the extraction of information and the 'assembly' of speech using emotional information disentangled from another speech sample rather than the original one. This 'arbitrary assembly' capability allows for training with multi-speaker and multi-emotion datasets.

Furthermore, since the speech information can be extracted and reassembled by the model, the trained model is not only capable of performing EVC (changing emotional information while preserving speaker and linguistic information) but also VC (changing speaker information) and even linguistic content conversion (changing linguistic information). The application of information disentanglement will create a comprehensive system capable of handling various speech-related generative tasks.

5.3.2 Exclusive Information Validation

After determining the system architecture, ensuring that each encoder extracts the corresponding information becomes essential. Two studies introduced an emotional classifier applied to the linguistic embedding to eliminate emotional information from the embedding [223, 224]. Inspired by this, a solution is proposed that not only eliminates unnecessary information but also preserves the required information, named Exclusive Information Validation (EIV).

Taking the emotional encoder as an example, its purpose is to extract output containing emotional information. Therefore, an emotional classifier is applied to supervise the emotional encoder using the following loss function:

$$\mathcal{L}_{EIV}^{emo \to emo} = -\sum_{i=1}^{C} y_i^{emo} \cdot log\, \hat{y}_i^{emo \to emo} \tag{5.12}$$

where the superscript $emo \to emo$ indicates that the loss is computed by using the extracted 'emotional' representation and the 'emotional' classifier. y^{emo} represents the 'emotional' label, and $\hat{y}^{emo \to emo}$ is the prediction of the 'emotional' classifier when fed with the extracted 'emotional' representation. And C is the number of classes.

Since the emotional representation is intended to be correctly recognised by the emotional classifier, the loss computation uses the classic cross-entropy loss. However, while the emotional classifier ensures that the representation contains emotional information, it does not guarantee the exclusive presence of emotional information. For instance, the emotional representation could still contain speaker information without affecting the classifier's performance. Therefore, a speaker classifier is implemented to eliminate speaker information using the MSE loss:

$$\mathcal{L}_{EIV}^{emo \to spk} = \frac{1}{C} \sum_{i=1}^{C} (\hat{y}_i^{emo \to spk} - \frac{1}{C})^2 \tag{5.13}$$

where C is the number of classes, and $\hat{y}^{emo \to spk}$ denotes the prediction of the 'speaker' classifier when fed with the extracted 'emotional' representation.

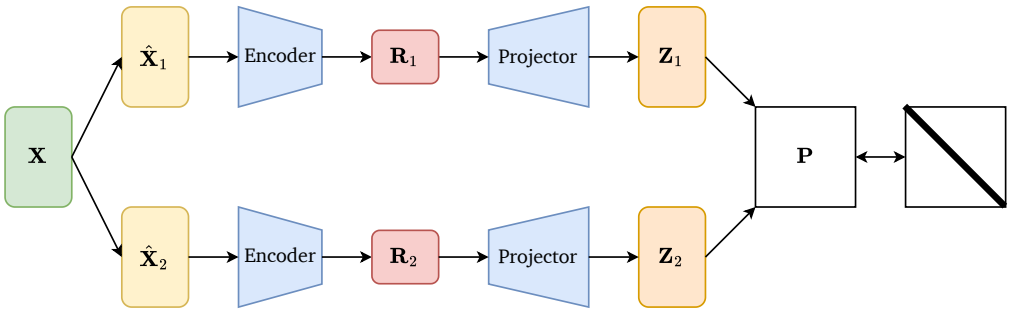
Since the goal is to eliminate any speaker information rather than preserving incorrect speaker information, the training target for $\hat{y}^{emo \to spk}$ is a uniform distribution $\left[\frac{1}{C}, \frac{1}{C}, \cdots, \frac{1}{C}\right]$, indicating that the classifier is 'confused' due to the absence of required information.

Thus, during the training phase, all three encoders and both classifiers are connected to guide the encoders in extracting the corresponding information. If the encoder and classifier types are the same, Equation 5.12 is used to compute the loss value. Otherwise, the loss function is Equation 5.13 to eliminate unnecessary information. The total EIV loss is computed as follows:

$$\begin{aligned} \mathcal{L}_{EIV} &= \lambda_{EIV}^{emo} \mathcal{L}_{EIV}^{emo} + \lambda_{EIV}^{spk} \mathcal{L}_{EIV}^{spk} \\ \mathcal{L}_{EIV}^{emo} &= \mathcal{L}_{EIV}^{emo \to emo} + \mathcal{L}_{EIV}^{spk \to emo} + \mathcal{L}_{EIV}^{lin \to emo} \\ \mathcal{L}_{EIV}^{spk} &= \mathcal{L}_{EIV}^{emo \to spk} + \mathcal{L}_{EIV}^{spk \to spk} + \mathcal{L}_{EIV}^{lin \to spk} \end{aligned} \tag{5.14}$$

where the superscript lin stands for 'linguistic'. λ_{EIV}^{emo} and λ_{EIV}^{spk} indicate the weights for the emotional and speaker classifiers, respectively. It is important to note that no classifier is implemented for linguistic information. The main reason is that a 'linguistic classifier' would be too large and complex for the current model. To achieve the same objectives as the other two classifiers, the 'linguistic classifier'

Figure 5.16: Architectural Framework of Barlow Twins



would need to be an ASR module, which is a sequential classification task. Even if a pre-trained ASR module were applied as the 'linguistic classifier', the output would be a large uniform distribution matrix rather than a vector, adding to the computational load. Therefore, the model is determined to use three encoders and two classifiers.

5.3.3 Correlation Consistency Validation

The application of EIV ensures that the output of a specific encoder contains only the intended information. However, the consistency of outputs from the same encoder, particularly the linguistic encoder that lacks a corresponding classifier, is also crucial for effective information disentanglement. To address this, Correlation Consistency Validation (CCV) is introduced to the model, which is explained in this section.

Consider a pair of speech samples $[E, S, L]$ and $[E, S', L']$, where E, S, and L represent emotional, speaker and linguistic information, respectively. The emotional category is the only the same information between the two samples. Therefore, after feeding these samples into the emotional encoder, their outputs should exhibit a high correlation. This principle of high correlation is extended to all three encoders, ensuring that each type of information is extracted correctly and consistently.

The concept of CCV is inspired by Barlow Twins [254], a self-supervised network architecture designed to enhance learning with large, unlabelled image datasets. The architecture of Barlow Twins is depicted in Figure 5.16. Barlow Twins aids the model in learning the embedding of input samples by utilising distortions. Specifically, an input image is distorted into two images that are then processed by an encoder to extract representations. Subsequently, a projector generates two embeddings from these representations. Finally, the model performs matrix-matrix multiplication on the embeddings to compute their correlation, with the training target being an identity matrix. Notably, a weight is applied to the off-diagonal elements during correlation computation.

Given that $\mathbf{Z}_1$ and $\mathbf{Z}_2$ are the latent embeddings of these two distorted images, the cross-correlation matrix $\mathbf{P}$ is computed as:

$$\mathbf{P} = \mathbf{Z}_1 \mathbf{Z}_2 \tag{5.15}$$

The loss during training is computed using the cross-correlation matrix and an identity matrix. For diagonal elements, the loss is computed as:

$$\mathcal{L}_{on\_diag} = \frac{1}{N} \sum_{i=1}^{N} (P_{ii} - 1)^2 \tag{5.16}$$

where N denotes the dimension of the cross-correlation matrix, and P_{ii} represents the element in the i-th row and the i-th column of $\mathbf{P}$. The loss for off-diagonal elements is:

$$\mathcal{L}_{off\_diag} = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} (P_{ij})^2 \tag{5.17}$$

Considering the weight applied to the off-diagonal loss:

$$\mathcal{L}_{BT} = \mathcal{L}_{on\_diag} + \lambda_{off\_diag} \mathcal{L}_{off\_diag} \tag{5.18}$$

where λ_{off_diag} represents the weight of the off-diagonal elements.

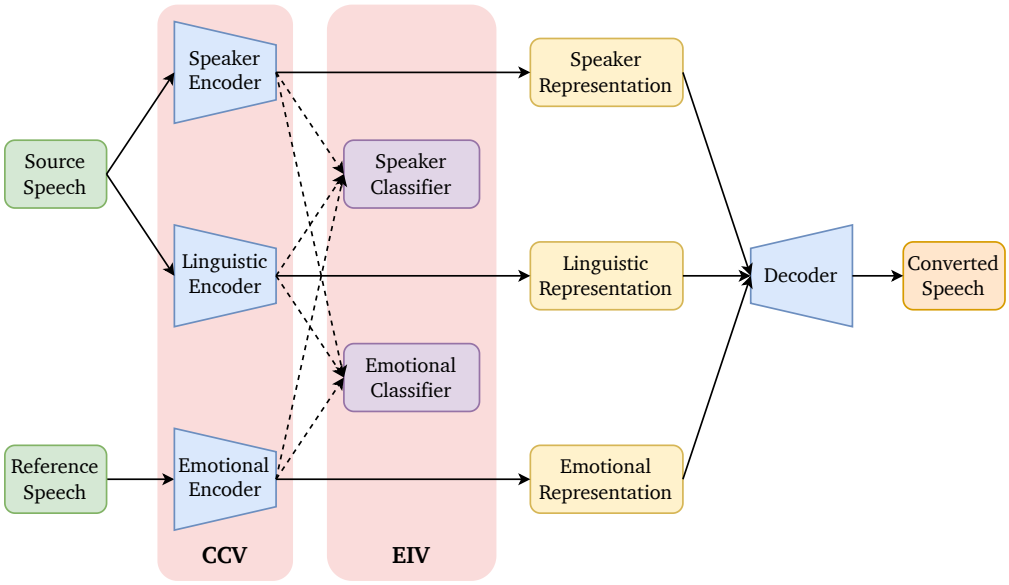
In the proposed scheme, a similar architecture to Barlow Twins is applied. Barlow Twins is trained to model the input image by understanding the similarity between two distortions. This concept can be adapted to understand the similarity between two speech samples. For example, CCV can learn the emotional information from $[E, S, L]$ and $[E, S', L']$ since the only sameness between them is the emotional expression.

Consequently, three auxiliary speech samples are required during training, with each sample sharing only one unique type of information with the original input according to the certain encoder. Through the combined supervision of EIV and CCV, the three encoders can effectively disentangle the information in speech for further processing.

5.3.4 Architectural Framework

Figure 5.17 illustrates the architectural framework of the proposed system. The system operates with two input speech samples: a source sample and a reference sample. Specifically, the source speech provides the speaker and linguistic information, while the reference speech supplies the emotional information. In terms of notation, given the source speech $[E, S, L]$ and the reference speech $[E', S', L']$, the model generates the converted speech $[E', S, L]$.

Figure 5.17: Illustration of the EVC System with Information Disentanglement



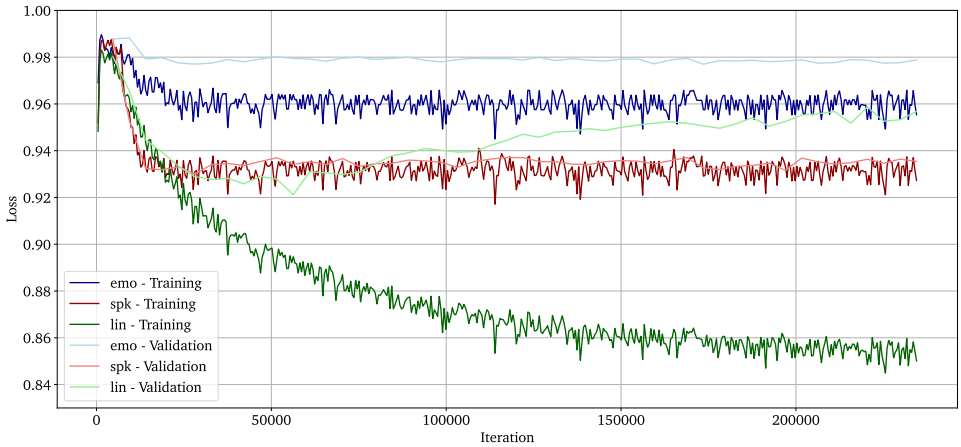
Within the system, the source speech is fed into both the speaker encoder and the linguistic encoder to extract the speaker representation and linguistic representation, respectively. Similarly, the emotional representation is extracted from the reference speech by the emotional encoder. Additionally, three auxiliary speech samples are provided to the encoders for CCV. Following this, the three extracted representations are fed into two classifiers for EIV. Finally, all three representations are passed to the decoder, which synthesises the converted speech.

The system builds upon the lighter Transformer architecture used in the experiments described in Section 5.2, with all three encoders sharing the same architecture. Each encoder consists of 4 encoder layers with 256 hidden units, 4 attention heads and a prenet—a linear layer that transforms the input Mel-spectrogram into a 256-dimensional embedding. Each encoder is followed by a projector composed of three linear layers that reduce the dimension of the representation from 256 to 20 for CCV, with $ReLU$ activation functions. For EIV, each classifier comprises three convolutional layers with $ReLU$ activation functions, followed by a linear layer.

5.3.5 Experimental Setup and Results

The ESD dataset has proven superior to the EmoV-DB dataset in terms of quality and training results, as discussed in Section 3.2. With the introduction of information disentanglement, it is now possible to utilise all English samples in the ESD dataset. The dataset comprises 350 groups of speech samples, with 300 groups

Figure 5.18: Loss Curves of Correlation Consistency Validation



(ID 51–350) designated as the training set, 20 groups (ID 1–20) as the validation set, and the remaining 30 groups (ID 21–50) as the test set. This equates to 15,000 samples in the training set, 1,000 samples in the validation set, and 1,500 samples in the test set. The expanded size of the training set, made possible by information disentanglement, offers significant advantages.

During the training phase, a source speech sample, denoted as $[E_S, S_S, L_S]$, is randomly selected from the training set. Two auxiliary samples, $[E'_S, S_S, L'_S]$ and $[E'_S, S'_S, L_S]$, are then randomly selected for CCV of the speaker encoder and linguistic encoder, respectively. For the reference speech, $[E_R, S_R, L_R]$ is randomly chosen, while the emotional auxiliary speech sample is $[E_R, S'_R, L'_R]$.

The loss function for the training process accounts for both EIV and CCV, and is expressed as:

$$\mathcal{L}_{ID} = \mathcal{L}_{EIV} + \mathcal{L}_{CCV} \tag{5.19}$$

$$\mathcal{L}_{CCV} = \lambda^{emo}_{CCV}\mathcal{L}^{emo}_{BT} + \lambda^{spk}_{CCV}\mathcal{L}^{spk}_{BT} + \lambda^{lin}_{CCV}\mathcal{L}^{lin}_{BT} \tag{5.20}$$

where all CCV losses ($\mathcal{L}_{CCV}$) are computed using Equation 5.18 with $\lambda = 0.05$. The weights for the emotional, speaker and linguistic encoders are represented by λ^{emo}_{CCV}, λ^{spk}_{CCV} and λ^{lin}_{CCV}, respectively. The optimiser used was Adam, with a learning rate of 1×10^{-5}, and the batch size was set to 32.

5.3.5.1 Correlation Consistency Validation

The first experiment aims to evaluate the performance of CCV. For this purpose, λ^{emo}_{CCV}, λ^{spk}_{CCV} and λ^{lin}_{CCV} are all set to 1, while λ^{emo}_{EIV} and λ^{spk}_{EIV} are set to 0. The results of the CCV are illustrated in Figure 5.18.

On the training set, the losses for the emotional and speaker encoders converge rapidly, reaching low levels after 24,000 iterations. However, the loss curve for the linguistic encoder flattens after approximately 200,000 iterations, likely due to the higher complexity of linguistic information. Additionally, all three encoders exhibit better performance on the validation set during training. A notable exception is the linguistic encoder, whose loss value decreases, reaching its lowest point at 52,680 iterations before gradually increasing. These results demonstrate that CCV effectively benefits all three encoders, enabling them to model the correlation between the source and auxiliary samples.

5.3.5.2 Full Experiment

The next step involves activating the training of the two classifiers, along with the decoder that combines the three representations into the converted Mel-spectrogram. Based on the results in Section 5.2.5, the lighter Transformer decoder, which includes 4 decoder layers and 4 heads, was integrated with the three encoders. The decoder takes the concatenated representations as input and generates the Mel-spectrogram autoregressively, assisted by a postnet. Additionally, a linear layer is applied at the end of the decoder to predict if the current frame is $\langle EOS \rangle$ token, which signals the termination of generation. Consequently, the MSE between the target and converted Mel-spectrograms is computed as the sequence-to-sequence loss. A stop loss is also incorporated, resulting in the following loss function:

$$\begin{aligned}\mathcal{L}_{ID-full} = &\lambda_{EIV}^{emo}\mathcal{L}_{EIV}^{emo} + \lambda_{EIV}^{spk}\mathcal{L}_{EIV}^{spk} \\ &+ \lambda_{CCV}^{emo}\mathcal{L}_{CCV}^{emo} + \lambda_{CCV}^{spk}\mathcal{L}_{CCV}^{spk} + \lambda_{CCV}^{lin}\mathcal{L}_{CCV}^{lin} \\ &+ \lambda_{S2S}\mathcal{L}_{S2S} + \lambda_{stop}\mathcal{L}_{stop}\end{aligned} \tag{5.21}$$

where λ_{S2S} and $\mathcal{L}_{S2S}$ represent the sequence-to-sequence weight and loss, respectively. Similarly, $\lambda stop$ and $\mathcal{L}_{stop}$ denote the weight and loss of the stop prediction. All weights in Equation 5.21 were set to 1, giving equal importance to all modules. Due to the additional modules, the batch size was adjusted from 32 to 20. The experimental results are presented in Figures 5.19, 5.20, 5.21 and 5.22.

Figure 5.19 illustrates the performance of CCV, with all curves following a similar pattern to those in Figure 5.18. However, all three training losses converge at higher levels, likely due to the added tasks. Consequently, the validation loss curves exhibit greater oscillation, especially in their later stages. These results confirm that CCV remains effective throughout the full experiment.

Regarding EIV, Figure 5.20 shows that both the emotional and speaker classifiers achieve excellent results, with losses nearing zero. Notably, the speaker classifier attains an outstanding cross-entropy loss of 0.025 on the validation set, while the emotional classifier performs slightly worse, with a minimum loss of 0.848. This

Figure 5.19: Loss Curves of Correlation Consistency Validation in Full Experiment

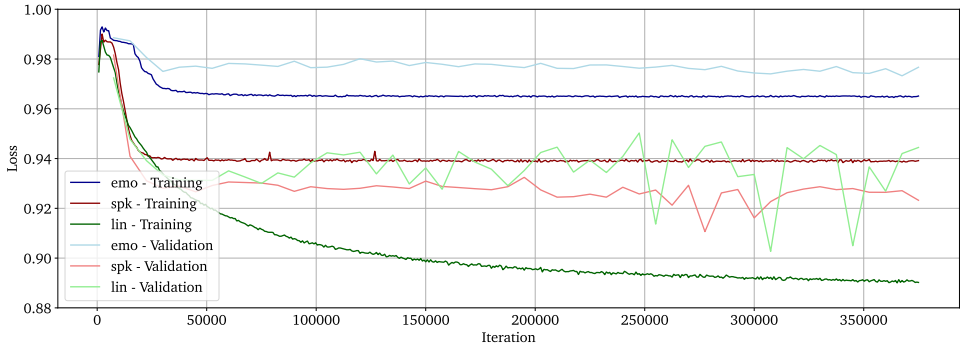


Figure 5.20: Loss Curves of Exclusive Information Validation in Full Experiment

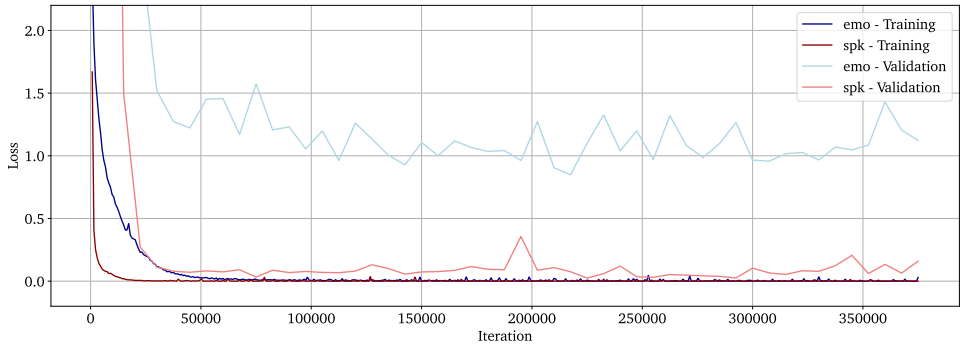


Figure 5.21: Loss Curves of Sequence-to-Sequence in Full Experiment

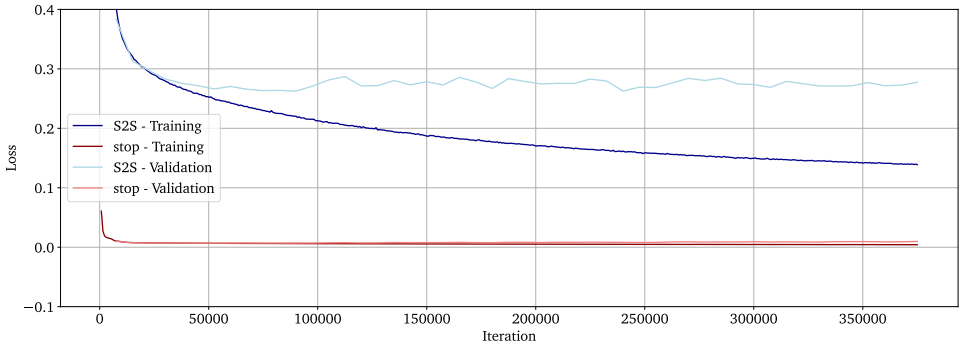
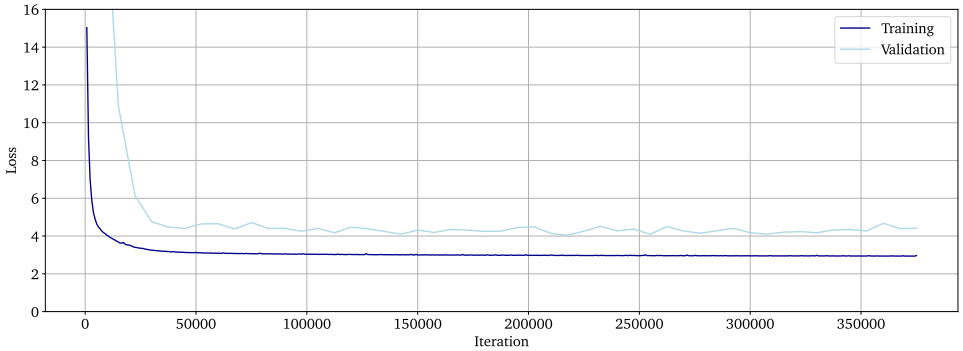


Figure 5.22: Loss Curves of Full Experiment



suggests that the differences in expression among speakers are more distinguishable to the classifiers than those among different emotional states.

As for the system's primary task, the sequence-to-sequence loss and stop loss are depicted in Figure 5.21. The EOS predictor not only learns effectively from the training set but also performs well on the validation set, leading to low loss values in both cases. The sequence-to-sequence training loss consistently decreases, showing a continuous decreasing trend even at the end of training. On the validation set, the MSE reaches 0.263 at $90,000$ iterations, but then increases with significant oscillation. This result is the best among all previous experiments, surpassing the last best validation loss of 0.295 reported in Section 5.2.2.

Figure 5.22 shows the total loss curves calculated using Equation 5.21. The training loss decreases smoothly over the iterations, while the validation loss follows a similar trajectory with higher oscillation, reaching a minimum of 4.042. This supports the excellent modelling and generative performance of the Transformer architecture. However, after reconstructing the speech using the Griffin-Lim algorithm, the converted speech exhibits clear speaker characteristics and emotional expression, though the linguistic content remains difficult to recognise.

This result does not imply that information disentanglement is ineffective. Figure 5.23 demonstrates the performance of information disentanglement using t-Distributed Stochastic Neighbour Embedding (t-SNE) [2]. t-SNE is a technique for visualising high-dimensional data in a two- or three-dimensional map [255], and it provides an intuitive way to assess whether an encoder preserves or eliminates certain information. To implement t-SNE, all test set samples were passed through the three encoders, and the output representations were processed by t-SNE. The dimension-reduced features generated by the same encoder were plotted in the same figure, with each point representing a speech sample. The colour of each point corresponds to the sample's label.

Figures 5.23a and 5.23b show the emotional representations coloured according to emotional and speaker labels, respectively. The representations are clearly clustered into five groups. In Figure 5.23a, the points within each group are mostly the same colour, with only a few outliers, indicating that the emotional encoder can extract similar representations for samples with the same emotional labels. Conversely, in Figure 5.23b, the coloured points are chaotically distributed, suggesting that the emotional encoder is not extracting similar representations from speech samples of the same speaker. In other words, the emotional encoder successfully removes speaker information from the input speech.

Figures 5.23c and 5.23d show similar results for the speaker encoder: samples with the same emotional expression are not grouped, while samples from the same speaker are successfully clustered. The last two figures indicate that the linguistic encoder excludes both emotional and speaker information, resulting in disordered colour distribution in both cases.

In conclusion, the t-SNE tests confirm that information disentanglement successfully preserves the desired information while eliminating other information in the emotional and speaker encoders. However, without a 'linguistic classifier', the linguistic encoder can only ensure the exclusion of emotional and speaker information, without guaranteeing the extraction of linguistic information. This is also reflected in the lack of intelligibility of the converted speech.

5.3.5.3 Optimisation of Information Disentanglement Weights

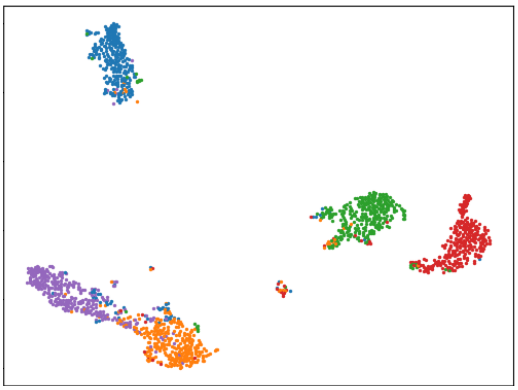
To optimise the intelligibility of the model, it is essential to enhance the linguistic encoder and ensure that the model focuses more on the target speech. Therefore, several experiments were conducted, varying the weights of the linguistic encoder and the sequence-to-sequence component to investigate their impact.

Table 5.3 presents a comparison of the results across eight different configurations, focusing on the weights λ_{CCV}^{lin} and λ_{S2S}. All other weights introduced in Equation 5.21 were held constant at 1. The first row of the table shows the results

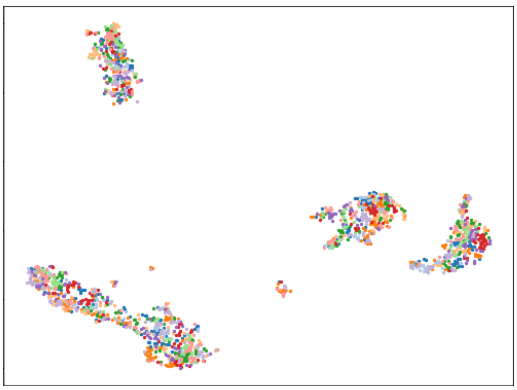
[2] `https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html`

Figure 5.23: Visualised Distribution of Representations by Using t-SNE

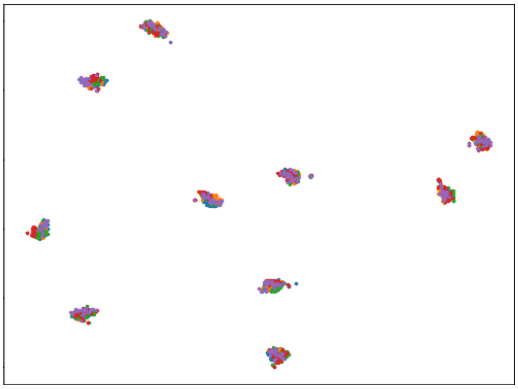
(a) ***emo*** Encoder + ***emo*** Labels



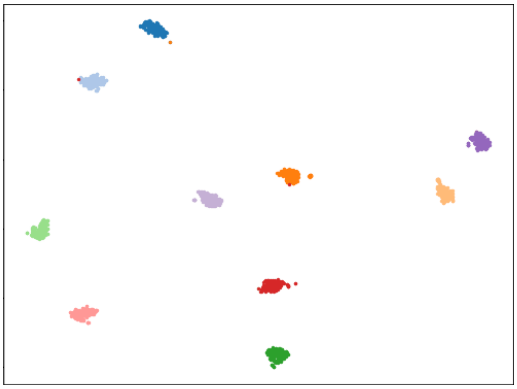
(b) ***emo*** Encoder + ***spk*** Labels



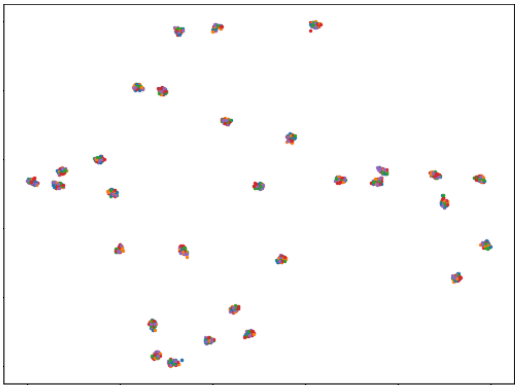
(c) ***spk*** Encoder + ***emo*** Labels



(d) ***spk*** Encoder + ***spk*** Labels



(e) ***lin*** Encoder + ***emo*** Labels



(f) ***lin*** Encoder + ***spk*** Labels

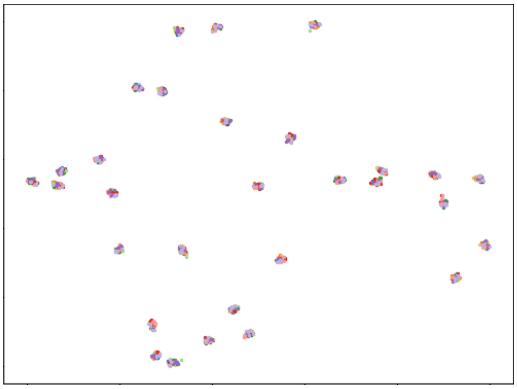


Table 5.3: Result Comparison among Different λ_{CCV}^{lin} and λ_{S2S}

λ_{CCV}^{lin}	λ_{S2S}	$\mathcal{L}_{EIV}$	$\mathcal{L}_{CCV}$	$\mathcal{L}_{S2S}$	$\mathcal{L}_{ID-full}$
1	1	0.923	2.799	0.263	4.042
1	2	0.894	**2.791**	0.258	4.009
1	5	0.922	2.825	0.261	4.052
1	10	0.977	2.832	**0.257**	4.099
2	5	**0.886**	2.805	0.259	**3.996**
5	5	0.924	2.841	0.260	4.045
10	5	0.965	2.833	0.262	4.088
10	10	0.983	2.837	0.262	4.099

of the previous full experiment. As demonstrated, no particular weight configuration stood out as significantly better than the others. For the sequence-to-sequence loss $\mathcal{L}_{S2S}$, which indicates the closeness of the converted speech to the target speech, the lowest value of 0.257 was achieved with $\lambda_{CCV}^{lin} = 1$ and $\lambda_{S2S} = 10$. However, this result does not show significant superiority over the worst value of 0.263, aside from being directly influenced by the higher sequence-to-sequence weight. Regarding the total loss $\mathcal{L}_{ID-full}$, the best result was obtained with $\lambda_{CCV}^{lin} = 2$ and $\lambda_{S2S} = 5$ at 3.996, but this was also not substantially different from the other results.

In conclusion, information disentanglement has been shown to improve the performance of sequence-to-sequence EVC, particularly in emotional expression and speaker characteristics. By leveraging information disentanglement, the range of the training set is extended, allowing for the use of multi-speaker, multi-emotion speech samples. The introduction of EIV and CCV to the model, through the application of two classifiers and three projectors, respectively, further enhances its performance. However, without a classifier to guide the extraction of linguistic information, the converted speech still lacks intelligibility. Therefore, integrating an ASR module as a 'linguistic classifier' is a practical solution. This classifier would help preserve linguistic information in the linguistic representation while excluding it from the other two representations. Given the complexity difference between a simple classifier and an ASR module, a pre-trained ASR module is more practical. Additionally, multi-task learning with TTS [187] and TTS assistant training [223, 224] are promising techniques to improve linguistic understanding.

6

Challenges, Outlook and Conclusion

To summarise, two distinct methods of computer-generated emotional expression, including ETTS and EVC, have been introduced, implemented, analysed and discussed. Beginning with a neutral TTS system, the research concentrated on addressing the challenge of emotionalising speech synthesis in scenarios where emotional speech datasets are limited. The investigation covered two ETTS systems based on transfer learning techniques, two GAN-based frame-to-frame EVC systems and two schemes for sequence-to-sequence EVC, all aiming to overcome the low-resource limitation from various perspectives. Each of these techniques and schemes represents a significant and viable direction in advancing HCI with artificial intelligence.

However, several challenges were identified during the research and implementation phases. This chapter introduces and discusses these challenges, along with the outlook they present, beginning with general challenges and then moving on to those specific to each method.

6.1 Challenges in Emotional Speech Synthesis

One primary challenge that has been repeatedly highlighted is the lack of training resources in ESS research, where the dataset is required to be clean, large and either text-speech parallel for ETTS or speech-speech parallel for EVC. Although alternative solutions, such as two-stage training [223], style pre-training [224], transfer learning (as proposed in Section 3.2) and even recent large models [253, 256] have been developed to alleviate the need for a large parallel emotional dataset, the main challenge for ESS remains the availability of suitable public datasets. Most existing emotional speech datasets are either too small or lacking in quality. For instance, the ESD dataset [99] is clean and parallel but contains only 350 groups

of samples per speaker. In contrast, the EmoV-DB dataset [100] has more groups of samples but includes non-speech utterances such as laughter (in *Amused*) and yawning (in *Sleepy*). Although an ETTS system was trained using EmoV-DB, a data cleansing process was necessary, which reduced the number of useful samples and required additional time and resources. Thus, the collection and release of appropriate datasets to the public would significantly advance research in this promising field and help optimise performance.

Another concern arises from the perspective of downstream application [8]. The application of ESS techniques is not limited to generating speech in a specific emotional category with a certain intensity, and so on. An ideal ESS scheme would allow users or downstream modules to adjust the nuanced emotional expression to their desired output, which requires effective disentanglement of mixed information in the speech [8].

The final challenge to discuss is related to comparison. Specifically, it is challenging to compare different ETTS and EVC models, respectively. Objective evaluation methods are not always intuitive, as they typically indicate how 'close' or 'similar' the output is to the target in specific evaluation dimensions [78]. However, synthetic speech that is considered as 'similar' by objective measures is not always perceived as such by humans. For example, the WER metric measures how many words are incorrectly pronounced in synthetic speech. This metric is generally interpreted as 'the lower, the better', aligning with human expectations for 'good speech'. Thus, comparing objective evaluation values across different models can be convincing to some extent.

However, only a few objective metrics align with human perception. For instance, the MCD metric measures the difference between the Mel-cepstrum of the output speech and the ground truth, but it does not reflect any aspects that humans focus on, such as emotional expression, speech quality, and naturalness.

Addressing this issue involves unifying objective and subjective evaluation metrics. Subjective metrics are more aligned with human standards, while objective evaluation is cheaper and faster. Promising approaches to mitigating this challenge include using evaluation methods based on pre-trained SER and ASR models or attempting to predict subjective evaluation scores, as targeted by the INTERSPEECH VoiceMOS Challenge 2022 [257]. However, until these methods mature sufficiently for practical use, subjective annotations will remain the gold standard for ESS evaluations.

Nevertheless, subjective evaluation also has drawbacks when comparing different models. One typical issue is participant bias between studies. For example, individuals from different cultural backgrounds may interpret emotional expressions in speech differently. This issue is aggravated when cultural differences involve different native languages.

Moreover, most ESS studies do not share the synthetic speech samples used in subjective evaluations, nor do they share the text of these samples. This makes it

impossible to ensure a fair comparison, given that context strongly influences how emotional expression is perceived by humans [258]. For instance, upon hearing a sentence like 'Great job.', people might assume it is *Happy*, even if it is spoken in a *Neutral* tone.

Finally, some studies are impossible to compare, even if they apply the same subjective evaluation criteria. For example, one study might use the MOS [222] to assess speech similarity in EVC research, while another might use BWS [224]. However, determining the 'best' or 'ideal' metric for the same criteria is difficult, as each metric has its own focus in evaluation. Additionally, investigating all available metrics individually would be highly labour-intensive.

In conclusion, the challenges in comparing models using both objective and subjective evaluation metrics are significant and pressing. As ESS advances, there will be a need for better solutions to replace simple evaluations of speech that are independent of context or interaction scenarios [78]. New qualitative evaluation methods will be required to identify fine-grained differences between ESS models, considering conversational contexts and speaker intents in more detail.

6.2 Outlook for Emotional Speech Synthesis

In recent years, the development of AI and neural networks has accelerated, leading to the rapid emergence of new models, architectures and demonstrations. In this chapter, several innovative techniques in TTS will be introduced, which have the potential to inspire further research in ESS.

6.2.1 Autoregressive and Non-Autoregressive Models

In statistics, an autoregressive model is defined as one that generates future outputs where the output can be expressed linearly by using past data points [259]. However, in generative models, the definition of an autoregressive model is broader and is not limited to linear relationships, as long as the generation process leverages previous information [22]. For example, the sequence-to-sequence ETTS models (Tacotron 2 and Transformer) introduced in Sections 3.2.4 and 3.2.6, as well as the Transformer-based EVC model in Section 5.2, can also be categorised as autoregressive models.

Recently, the term 'sequence-to-sequence' has become more commonly used to describe the task that the model performs, which is a transformation from one sequence to another sequence. Sequence-to-sequence models take one sequence as input and generate another sequence [39]. For instance, ETTS generates speech sequences from textual sequences, and EVC generates speech sequences from other speech sequences. This characteristic sets sequence-to-sequence models apart from autoregressive models.

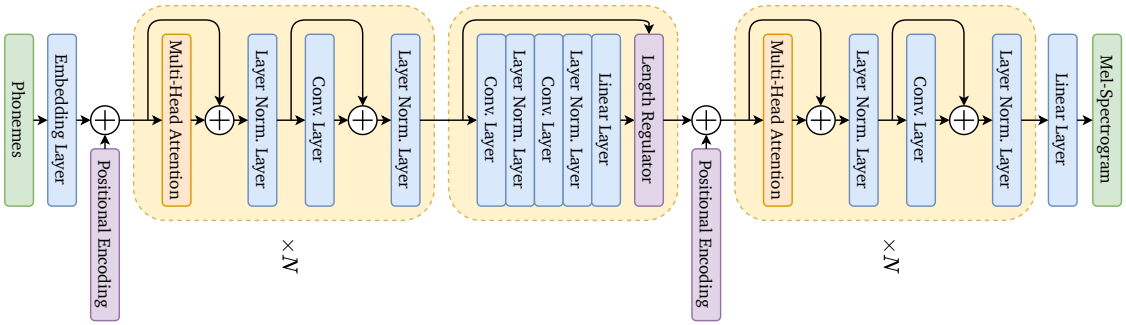
A good example of an autoregressive model that is not sequence-to-sequence is the Generative Pre-trained Transformer (GPT) series, such as GPT-3 [260]. GPT-3, without separate encoders and decoders, understands the input text and generates the output sequence autoregressively using a stack of Transformer decoders. GPT-3 is not considered a sequence-to-sequence model because no translation between domains is involved. Conversely, an example of a non-autoregressive sequence-to-sequence model is the TTS model FastSpeech [33].

The motivation for non-autoregressive solutions lies in addressing the limitations of autoregressive models. Firstly, due to the inherent nature of autoregression, generating one frame requires the completion of all previous frames. Although teacher-forced learning speeds up processing during training, it still results in long inference times, even for CNN and Transformer-based models [32, 40]. Additionally, errors generated at one timestep accumulate across all subsequent timesteps, affecting output speech quality with issues like word-skipping and repeating [32]. Another challenge with autoregressive TTS models is the limited controllability of speech rate and prosody, which are crucial for emotional expressivity [33]. This is because autoregressive models typically use trainable attention modules for text-speech alignment, which is not an explicit solution.

FastSpeech [33] addresses these challenges through a novel architecture, shown in Figure 6.1. Instead of the conventional encoder-decoder architecture used in Transformers [43], FastSpeech employs a feed-forward architecture to achieve parallel generation, significantly increasing training speed. The model uses a multi-head self-attention mechanism and 1D convolutional networks to learn cross-position information and adjacent relationships, respectively. Rather than relying on the encoder-decoder attention module for alignment, FastSpeech uses a phoneme duration predictor for hard alignment, acquired from a pre-trained autoregressive Transformer TTS model. This design reduces error accumulation and poor alignment, mitigating issues like word-skipping and repeating. Furthermore, whereas conventional autoregressive models determine generation termination based on the current generated frame, such as recognising an $\langle EOS \rangle$ token [39] or predicting a stop signal [40], FastSpeech uses a length regulator. This regulator leverages the duration of each phoneme in the sentence to determine the output's length.

Building on FastSpeech, FastSpeech 2 [261] introduced improvements in two key areas. Forced alignment replaced pre-trained autoregressive text-speech alignment, and the model introduced control over speech duration, pitch and energy through conditional inputs. Moreover, FastSpeech 2s [261] replaces the Mel-spectrogram decoder with a waveform decoder, creating an end-to-end TTS system. Given the strong performance of the FastSpeech series, their potential in ETTS has been explored through various designs [262, 263].

Figure 6.1: Architectural Framework of FastSpeech



6.2.2 Flow-based TTS

Deterministic models, such as Autoencoders, generate a unique output that directly depends on the input. This means that once the model and its input are defined, the output remains consistent. However, this deterministic nature limits the creativity of such models in generating speech, as the objective of TTS is not merely to produce identical speech outputs but to generate human-like, natural variations. Therefore, models based on probability distributions are better suited for generative tasks.

One probabilistic model, the VAE, has been introduced in Section 4.2.4.1. Another common probabilistic model used in TTS is the Flow-based model. Unlike the VAE, where the encoder creates a Gaussian distribution to sample a vector for the decoder to generate the output, the flow-based model, also known as a normalising flow, uses a series of invertible functions to convert a Gaussian distribution directly into the output [264]. A prominent example of a flow-based TTS model is Parallel WaveNet [63]. The original WaveNet model discarded the recurrent architecture in favour of dilated causal convolutional layers, which increased processing speed while maintaining long-range temporal dependency due to the larger receptive field [54]. However, the generation speed was still slow because the process was inherently sequential and autoregressive. Parallel WaveNet improved processing speed by introducing the inverse autoregressive flow [265], which constructs a multivariate distribution as an invertible non-linear function [63]. This allowed the sampling process to be parallelised, significantly reducing processing time.

Beyond autoregressive approaches, bipartite transformation has also been explored in flow-based TTS research. Since the transformation must be invertible, affine coupling layers—incorporating operations like splitting, concatenation and permutation—were used to ensure that each dimension affects the others effectively [266]. WaveGlow [55] implemented bipartite transformation in the TTS task, achieving better speech quality than WaveNet in the MOS test (3.96 ± 0.13 to 3.89 ± 0.12) with a faster inference speed than Parallel WaveNet. FloWaveNet [267]

also utilised affine coupling layers, focusing on eliminating the need for two-stage training in Parallel WaveNet by using only a single maximum likelihood loss.

6.2.3 Diffusion-based TTS

The diffusion model is one of the most popular and powerful deep learning models recently, and it has been widely applied to various generative tasks, including text, audio, speech, image and video generation [268]. Denoising Diffusion Probabilistic Model (DDPM), Noise Conditioned Score Network and Stochastic Differential Equation are the three main architectures of diffusion models. These models aim to find a mapping between the original sample and a probability distribution, typically a Gaussian distribution [269].

Taking DDPMs as an example [270], during the training phase, random Gaussian noise is iteratively added to the original sample until it becomes pure Gaussian noise. This forward process can be expressed as:

$$\mathbf{x}_t = \sqrt{1-\beta_t}\,\mathbf{x}_{t-1} + \sqrt{\beta_t}\,\mathbf{z}_{t-1} \tag{6.1}$$

where $\mathbf{x}_0$ is the original sample, $\mathbf{x}_t$ represents the corrupted sample after noise $\mathbf{z}$ has been added for t times. The different subscripts of $\mathbf{z}_t$ and $\mathbf{z}_{t-1}$ means these two noises are sampled from $\mathcal{N}(0, \mathbf{I})$ independently, where $\mathbf{I}$ represents the identity matrix. $[\beta_1, \beta_2 \cdots \beta_t]$ denote the series of hyperparameters controlling the intensity of the noise. To simplify this process, another coefficient $\bar{\alpha}_t$ is introduced:

$$\bar{\alpha}_t = \prod_{n=1}^{t}(1-\beta_n) \tag{6.2}$$

which leads to an equation that can compute the corrupted sample from the original sample directly by performing a single sampling of the noise:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\mathbf{z}_0 \tag{6.3}$$

where $\mathbf{z}_0$ denotes the sampled noise from $\mathcal{N}(0, \mathbf{I})$. The goal of forward process is to train a noise predictor $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ which is used in the inference phase to predict the noise from the corrupted sample $\mathbf{x}_t$ and the current denoising step t.

During the inference phase, given a Gaussian noise sample $\mathbf{x}_t$, DDPMs generate the denoised output step by step. This process, known as the reverse process, is expressed as:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\,\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}_0 \tag{6.4}$$

where the coefficients

$$\alpha_t = 1 - \beta_t \tag{6.5}$$

$$\sigma_t = \sqrt{\beta_t} \text{ or } \sigma_t = \sqrt{\frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t}\beta_t} \tag{6.6}$$

The reverse process is repeated t times until $\mathbf{x}_0$ is computed, resulting in the final denoised sample. Diffusion models have demonstrated outstanding performance across various AIGC tasks [268]. However, they also have notable limitations, primarily from the multiple diffusion steps required during generation, which results in slower inference speeds compared to other probabilistic generative models [269].

Diffusion models have also achieved remarkable results in TTS research, as evidenced by models like Diff-TTS [48], Grad-TTS [47] and Guided-TTS [271, 272]. Diff-TTS introduced the diffusion architecture to restore noise to the Mel-spectrogram conditioned on the input text and employed accelerated sampling to improve inference speed [48]. The unaccelerated Diff-TTS achieved speech quality comparable to the reconstructed ground truth (extracting Mel-spectrogram and then reconstructing it with HiFi-GAN) in the MOS test, scoring 4.34 ± 0.06 compared to 4.32 ± 0.06. Although the accelerated sampling implementation decreased speech quality slightly, it significantly improved inference RTF, reducing it from 1.744 to as low as 0.035 [48].

6.2.4 Large Models and TTS

The GPT series, developed by OpenAI, represents a significant advancement in the field of NLP. Starting from GPT-1 [273] with 110 million parameters to GPT-2 with 1.5 billion parameters [274], the performance of LLMs has been validated across various NLP tasks. In 2018, Google introduced a Transformer-based neural network called Bidirectional Encoder Representations from Transformers (BERT), designed for tasks like Masked Language Modelling and Next Sentence Prediction [243]. BERT's base model, with 110 million parameters, and its larger variant, with 340 million parameters, delivered excellent results. Subsequently, OpenAI released GPT-3 [260], which utilised an astounding 175 billion parameters. Inspired by the success of models with vast numbers of parameters, recent LLMs have seen exponential growth in size, with GPT-4 reportedly reaching 4 trillion parameters [244].

Researchers have also explored the application of large models in audio processing. A common approach in voicebot implementations involves concatenating ASR, LLM and TTS systems to leverage the capabilities of LLMs, such as those in ChatGPT 4 [275]. However, inspired by the exceptional performance of LLMs in text-based tasks, audio-based large models have emerged. For instance, the BERT-like approach of predicting masked tokens was adapted for speech

representation learning, resulting in Hidden-unit BERT (HuBERT) [276]. HuBERT introduced three models of varying sizes, with the largest containing 1 billion parameters, improving the prediction of predetermined cluster assignments. The extracted representations from HuBERT have been effectively used in EVC tasks by combining them with speaker and emotion-label representations [253].

Additionally, contrastive learning has been applied to speech representation extraction, leading to another BERT-based model called w2v-BERT [277]. The pre-training of w2v-BERT involves multitasking, training on both masked language modelling and contrastive tasks simultaneously. w2v-BERT is also considered a large model for audio, with its largest variant, w2v-BERT XXL, employing 1 billion parameters. The w2v-BERT XL, which includes 0.6 billion parameters, was utilised in AudioLM [278] to acquire discrete semantic tokens for audio generation.

Similarly, the TTS model VALL-E [256] implemented a strategy using discrete codes instead of continuous ones, both from input acoustic prompts and phonemes. This approach transforms TTS from a signal regression task into a conditional NLP task. VALL-E's neural audio codec enables the synthesis of personalised speech using only three seconds of an unseen speaker's speech in a zero-shot condition. To enhance performance in multilingual tasks such as cross-lingual TTS and speech-to-speech translation, an improved version called VALL-E X [279] was introduced.

Despite the impressive performance of large models in speech-related tasks, their computational requirements are substantial. For example, VALL-E required 16 NVIDIA® Tesla® V100 32GB graphics cards for $800,000$ training steps [256], while VALL-E X increased the number of graphics cards to 32 [279]. Even for speech audio extractors like HuBERT BASE, 32 graphics cards were used for $650,000$ training steps, while HuBERT LARGE and HuBERT X-LARGE required 128 and graphics cards, respectively [276]. These requirements limit the accessibility of this research field, particularly for researchers and students in academic settings. As a result, studies focusing on the application of pre-trained large models are more practical, such as using the pre-trained models without modifications [253] or employing Parameter-Efficient Fine-Tuning [280, 281].

Moreover, training large models necessitates an extensive corpus, which is often not feasible in emotional speech research as discussed in Section 3.1.3. For instance, HuBERT BASE was trained on 960 hours of speech from LibriSpeech, while HuBERT LARGE and HuBERT X-LARGE used $60,000$ hours from Libri-light [276]. VALL-E utilised both LibriSpeech and Libri-light [256], while VALL-E X employed an additional 73 million sentence pairs for speech-to-speech translation training [279]. However, benefiting from the extensive training data and complex model architecture, VALL-E and VALL-E X inherently preserve the emotional expression in the audio prompt [256, 279]. Fine-tuning a pre-trained large model for audio with a small emotional speech dataset is also a viable approach.

In conclusion, while there are significant challenges and limitations in current research, advancements in technology, corpora and evaluation metrics will continue

to improve the performance of ESS. In this thesis, ESS is divided into two research areas: ETTS and EVC, based on their respective application scenarios. The background, state-of-the-art studies and key concepts of both fields are explored, followed by the implementation and analysis of several models. For ETTS, one neutral TTS system and two ETTS systems were designed and tested. Additionally, two frame-to-frame EVC systems and two sequence-to-sequence EVC systems were developed. Experimental results demonstrate that ESS models can be implemented to generate emotional speech; however, further optimisation is necessary to achieve better, more natural and higher-quality performance. This thesis contributes to the ongoing exploration of ESS and paves the way for future research to address the challenges identified, ultimately aiming to improve human-computer interactions through more sophisticated and emotionally resonant speech synthesis technologies. Nevertheless, ESS remains a promising research field and is expected to play a crucial role in HCI, enhancing efficiency in work, study and daily life.

Acronyms

ABX ..... ABX Test (identifies X as either A or B)
AI ..... Artificial Intelligence
AIGC ..... Artificial Intelligence Generative Content
ASR ..... Automatic Speech Recognition
BAPD ..... Band Aperiodicity Distortion
BFCC ..... Bark-Frequency Cepstral Coefficient
BWS ..... Best-Worst Scaling
CA ..... Computer Audition
CCV ..... Correlation Consistency Validation
CER ..... Character Error Rate
CMU-Dict ..... CMU Pronouncing Dictionary
CNN ..... Convolutional Neural Network
CTC ..... Connectionist Temporal Classification
CV ..... Computer Vision
CWT ..... Continuous Wavelet Transformation
DBN ..... Deep Belief Network
DCT ..... Discrete Cosine Transformation
DDPM ..... Denoising Diffusion Probabilistic Model
DDUR ..... Difference of Duration
DMOS ..... Degradation Mean Opinion Score
DNN ..... Deep Neural Network

DTW ..... Dynamic Time Warping
DWT ..... Discrete Wavelet Transformation
EIV ..... Exclusive Information Validation
EOS ..... End of the Sequence
ESS ..... Expressive/Emotional Speech Synthesis
ETTS ..... Expressive/Emotional Text-to-Speech
EVC ..... Emotional Voice Conversion
FD ..... Frobenius Distance
FFE ..... F_0 Frame Error
FFT ..... Fast Fourier Transformation
FT ..... Fourier Transformation
GAN ..... Generative Adversarial Network
GMM ..... Gaussian Mixture Model
GPE ..... Gross Pitch Error
GRU ..... Gated Recurrent Unit
HCI ..... Human-Computing Interaction
HMM ..... Hidden Markov Model
IFT ..... Inverse Fourier Transformation
KLD ..... Kullback-Leibler Divergence
LDA ..... Linear Discriminative Analysis
LLM ..... Large Language Model
LPC ..... Linear Predictive Coding
LSD ..... Log-Spectral Distortion
LSP ..... Line Spectral Pair
LSTM ..... Long Short-Term Memory
MCC ..... Mel-Cepstral Coefficient
MCD ..... Mel-Cepstral Distortion
MCEP ..... Mel-Cepstral Coefficient
MFCC ..... Mel-Frequency Cepstral Coefficient
MLP ..... Multi-Layer Perceptron

MOS . . . . . Mean Opinion Score
MSE . . . . . Mean Squared Error
NLP . . . . . Natural Language Processing
PCM . . . . . Pulse-Code Modulation
POS . . . . . Part of Speech
PPG . . . . . Phonetic Posteriorgram
PT . . . . . Preference Test
RBM . . . . . Restricted Boltzmann Machine
RMSE . . . . . Root Mean Square Error
RNN . . . . . Recurrent Neural Network
RTF . . . . . Real-Time Factor
SER . . . . . Speech Emotion Recognition
SNR . . . . . Signal-to-Noise Ratio
SSE . . . . . Smoothed Spectral Envelope
STFT . . . . . Short-Time Fourier Transformation
SOS . . . . . Start of the Sequence
SPSS . . . . . Statistical Parametric Speech Synthesis
TTS . . . . . Text-to-Speech
t-SNE . . . . . t-Distributed Stochastic Neighbour Embedding
UAR . . . . . Unweighted Average Recall
VAE . . . . . Variational Autoencoder
VAE-GAN . . . . . Variational Autoencoder Generative Adversarial Network
VAW-GAN . . . . . Variational Autoencoder Wasserstein Generative Adversarial Network
VC . . . . . Voice Conversion
VDE . . . . . Voicing Decision Error
WER . . . . . Word Error Rate

List of Symbols

Content-based Sequence-to-Sequence Learning

$\mathbf{X}$ . . . . . Input sequence

x_t . . . . . t-th frame/token of input sequence

$\mathbf{M}$ . . . . . Memory sequence

m_t . . . . . t-th frame/token of memory sequence

T . . . . . Length of input sequence

$enc(\cdot)$ . . . . . Encoding process

$\mathbf{C}$ . . . . . Context vector

c_t . . . . . t-th frame/token of context vector

α_{ti} . . . . . Content-based attention weight

$exp(\cdot)$ . . . . . Exponential function

$aln(\cdot)$ . . . . . Alignment function

$\hat{y}_t$ . . . . . t-th frame/token of predicted sequence

y_t . . . . . t-th frame/token of ground truth target

Signal Processing and Feature Extraction

$\hat{f}(\cdot)$ . . . . . Spectrum

ξ . . . . . Frequency variable

$exp(\cdot)$ . . . . . Exponential function

$f(\cdot)$ . . . . . Waveform

t . . . . . Time variable
i . . . . . Imaginary unit
$w(\cdot)$ . . . . . Window function
n . . . . . Index of the sliding window
h . . . . . Hop length of the sliding window
l . . . . . Length of the sliding window
S . . . . . Spectrogram
$Mel(\cdot)$ . . . . . Mel-scale
$W(\cdot)$ . . . . . Continuous Wavelet Transformation
s . . . . . Scale parameter
p . . . . . Position parameter
$F_0(\cdot)$ . . . . . Fundamental frequency
$\psi(\cdot)$ . . . . . Mother wavelet function
$\psi_M(\cdot)$ . . . . . Mexican hat wavelet function
$\tilde{W}(\cdot)$ . . . . . Discrete Wavelet Transformation
J . . . . . Number of discrete scales
k . . . . . Translation parameter
E_t . . . . . Energy of spectrum at t-th frame
D . . . . . Dimension of spectrum
T . . . . . Number of spectrum frames
$\hat{F}_0$ . . . . . Normalised fundamental frequency
μ_s, μ_t . . . . . Means across all source and target speech samples
σ_s, σ_t . . . . . Standard deviation across all source and target speech samples

Evaluation Metrics

$\mathbf{Y}$ . . . . . Target Sequence
y_t . . . . . t-th element of target sequence
$\hat{\mathbf{Y}}$ . . . . . Synthetic Sequence
$\hat{y}_t$ . . . . . t-th element of synthetic sequence

T . . . . . Length of sequences

$\mathbf{M}$ . . . . . Target MFCCs

$m_{t,n}$ . . . . . Element of target MFCCs at t-th frame and n-th dimension

$\hat{\mathbf{M}}$ . . . . . Synthetic MFCCs

$\hat{m}_{t,n}$ . . . . . Element of synthetic MFCCs at t-th frame and n-th dimension

N . . . . . Number of dimensions

$\mathbf{V}$ . . . . . Target voice decision sequence

v_t . . . . . Voice decision of t-th frame in target speech

$\hat{\mathbf{V}}$ . . . . . Synthetic voice decision sequence

$\hat{v}_t$ . . . . . Voice decision of t-th frame in synthetic speech

Loss Functions

$\mathcal{L}_{\mathcal{T}-\mathcal{TTS}}$ . . . . . Loss function of Transformer TTS

$\mathcal{L}_{mel}$ . . . . . Loss function of the decoder output

$\mathcal{L}_{post}$ . . . . . Loss function of the postnet output

$\mathcal{L}_{stop}$ . . . . . Loss function of stop token prediction

λ_{stop} . . . . . Weight of stop token prediction

$\mathcal{L}_{adv}(\cdot)$ . . . . . Adversarial loss function

$G_{X\to Y}$ . . . . . Generator that takes X and generate Y

D_Y . . . . . Discriminator that discriminate Y

$\mathcal{L}_{cyc}(\cdot)$ . . . . . Cycle consistency loss function

λ_{cyc} . . . . . Weight of cycle consistency loss

$\mathcal{L}_{id}(\cdot)$ . . . . . Identity loss function

λ_{id} . . . . . Weight of identity loss

$\mathcal{L}_{CycleGAN}$ . . . . . Loss function of CycleGAN

$\mathcal{L}_{KL}(\cdot)$ . . . . . Kullback-Leibler loss function

λ_{KL} . . . . . Weight of Kullback-Leibler loss

$D_{KL}(A||B)$ . . . . . Kullback-Leibler distance between A and B

$\mathcal{L}_{recon}$ . . . . . Reconstruction loss function

λ_{recon} . . . . . Weight of reconstruction loss

$\mathcal{L}_E$ . . . . . Encoder loss function

λ_{adv} . . . . . Weight of adversarial loss

$\mathcal{L}_G$ . . . . . Generator loss function

$\mathcal{L}_D$ . . . . . Discriminator loss function

$\mathcal{L}_{VAE-GAN}(\cdot)$ . . . Loss function of VAE-GAN

$\mathcal{L}_{EIV}$ . . . . . Exclusive information validation loss function

$^{emo \to spk}$ . . . . . Output of *Emotional* encoder to *Speaker* classifier

y_i^{emo} . . . . . *Emotional* label of i-th speech sample

$\hat{y}_i$ . . . . . Predicted label of i-th generated speech sample

C . . . . . Number of classes

λ_{EIV}^{emo} . . . . . Weight of *Emotional* exclusive information validation loss

$\mathbf{P}$ . . . . . Cross-correlation matrix

$\mathbf{Z}_1, \mathbf{Z}_2$ . . . . . Latent embeddings by distorting the same image

$\mathcal{L}_{on\_diag}$ . . . . . Loss function of diagonal elements

N . . . . . Dimension of cross-correlation matrix

P_{ij} . . . . . The element in i-th row and j-th column of matrix $\mathbf{P}$

$\mathcal{L}_{off\_diag}$ . . . . . Loss function of off-diagonal elements

$\mathcal{L}_{CCV}$ . . . . . Correlation consistency validation loss function

$\mathcal{L}_{BT}$ . . . . . Loss function of Barlow Twins

λoff_diag . . . . . Weight of off-diagonal loss

$\mathcal{L}_{ID}$ . . . . . Loss function of information disentanglement

$\mathcal{L}_{BT}^{emo}$ . . . . . Barlow Twins loss of *Emotional* encoder

λ_{CCV}^{emo} . . . . . Weight of *Emotional* correlation consistency validation loss

$\mathcal{L}_{S2S}$ . . . . . Sequence-to-sequence loss function

λ_{S2S} . . . . . Weight of sequence-to-sequence loss

Neural Networks

x . . . . . Input

y . . . . . Output

$a(\cdot)$ . . . . . Activation function

w_i . . . . . Weight of i-th neuron

b . . . . . Bias

$P(A|B)$ . . . . . Probability of A under the condition B

$sigmoid(\cdot)$ . . . . . Sigmoid function

v_i . . . . . i-th visible neuron

h_j . . . . . j-th hidden neuron

J . . . . . Number of hidden neurons

w_{ij} . . . . . Weight between neuron i and neuron j

b_i . . . . . Bias of neuron i

y_t . . . . . t-th element of output

f_t . . . . . Forget gate of t-th cell

i_t . . . . . Input gate of t-th cell

$\tilde{c}_t$ . . . . . Update of t-th cell

c_t . . . . . State of t-th cell

o_t . . . . . Output gate of t-th cell

$w_{\overrightarrow{h}y}$ . . . . . Forward weight between hidden state and output

$w_{\overleftarrow{h}y}$ . . . . . Backward weight between hidden state and output

$\overrightarrow{h}_t$ . . . . . Forward hidden state of t-th cell

$\overleftarrow{h}_t$ . . . . . Backward hidden state of t-th cell

$enc(\cdot)$ . . . . . Encoding process

$dec(\cdot)$ . . . . . Decoding process

z . . . . . Latent representation

x' . . . . . Reconstructed input

$q(z|x)$ . . . . . Probability distribution of z given x

$p(x|z)$ . . . . . Probability distribution of x given z

ϕ, θ . . . . . Parameters of distribution

$\mathcal{N}(z|a,b)$ . . . . . Gaussian probability distribution of z, constructed by mean a and variance b

$\mu(\cdot)$ . . . . . Mean

$\sigma(\cdot)$ . . . . . Standard deviation

$\underset{A}{min}B$ . . . . . The minimum of B with respect to A

$\underset{A}{max}B$ . . . . . The maximum of B with respect to A

$V(\cdot)$ . . . . . Value function

G . . . . . Generator

D . . . . . Discriminator

$\mathbb{E}_A B$ . . . . . Expectation of B with respect to A

$\boldsymbol{x} \sim p(\boldsymbol{x})$ . . . . . $\boldsymbol{x}$ is sampled from probability distribution $p(\boldsymbol{x})$

$\hat{x}$ . . . . . Normalised input

$min(\cdot)$ . . . . . Minimum value

$max(\cdot)$ . . . . . Maximum value

Activation Functions

x . . . . . Activation function input

$tanh(\cdot)$ . . . . . Hyperbolic tangent function

$exp(\cdot)$ . . . . . Exponential function

$softmax(\cdot)$ . . . . . Softmax function

Transformer

$\mathbf{X}, \mathbf{X}'$ . . . . . Sequences

x_i, x_i' . . . . . i-th element in $\mathbf{X}$ and $\mathbf{X}'$

k_i . . . . . i-th element of key matrix

W_K . . . . . Parameter matrix of key

v_i . . . . . i-th element of value matrix

W_V . . . . . Parameter matrix of Value

q_i . . . . . i-th element of query matrix

W_Q . . . . . Parameter matrix of Query

α . . . . . Weight vector
K . . . . . Key matrix
K^T . . . . . Transpose of key matrix
d_K . . . . . Dimension of key matrix
C . . . . . Context matrix
c_i . . . . . i-th element of context matrix
V . . . . . Value matrix
$attn_{sdp}(\cdot)$ . . . . . Scaled dot-product attention function
Q . . . . . Query matrix
$\mathbf{S}$ . . . . . Source sequence
$\mathbf{T}$ . . . . . Target sequence
$\hat{\mathbf{T}}$ . . . . . Generated sequence
$multihead(\cdot)$ . . . Multihead attention function
W_O . . . . . Multihead weight parameter matrix
$head_i$ . . . . . i-th attention head
h . . . . . Number of attention heads
$pe(\cdot)$ . . . . . Positional encoding
$sin(\cdot)$ . . . . . Sine trigonometric functions
$cos(\cdot)$ . . . . . Cosine trigonometric functions
p . . . . . Position index
i . . . . . Dimension index
d . . . . . Dimension of model

Training Strategies

y_n . . . . . n-th element of ground truth sequence
$\hat{y}_n$ . . . . . n-th element of generated sequence
$model(\cdot)$ . . . . . Generative model
ϵ . . . . . Proportion of ground truth frames
$max(\cdot)$ . . . . . Maximum value

t . . . . . t-th epoch in training phase

$\lfloor a/b \rfloor$ . . . . . Integer division of a by b

Diffusion Models

$\mathbf{x}_t$ . . . . . Corrupted sample after added noise for t times

β . . . . . Noise intensity hyperparameter

$\mathbf{z}$ . . . . . Noise sampled from Gaussian distribution

$\boldsymbol{\epsilon}_\theta(\cdot)$ . . . . . Noise predictor with parameter set θ

$\bar{\alpha}, \alpha, \sigma$ . . . . . Noise intensity coefficients

Bibliography

[1] J. M. Carroll, "Human-computer interaction: Psychology as a science of design," *International Journal of Human-Computer Studies*, vol. 46, no. 4, pp. 501–522, 1997.

[2] Y. D. Kataware and U. L. Bombale, "A wearable wireless device for effective human computer interaction," *International Journal of Computer Applications*, vol. 99, no. 9, pp. 9–14, 2014.

[3] Z. Lv, F. Poiesi, Q. Dong, J. Lloret, and H. Song, "Deep learning for intelligent human-computer interaction," *Applied Sciences*, vol. 12, no. 22, p. 11457, 2022.

[4] N. J. Nilsson, *The Quest for Artificial Intelligence*. Cambridge University Press, 2009.

[5] C. Ittichaichareon, S. Suksri, and T. Yingthawornsuk, "Speech recognition using MFCC," in *Proc. ICGSM*, Pattaya, Thailand, 2012, pp. 135–138.

[6] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J. G. Taylor, "Emotion recognition in human-computer interaction," *IEEE Signal Processing Magazine*, vol. 18, no. 1, pp. 32–80, 2001.

[7] B. W. Schuller, G. Rigoll, and M. Lang, "Speech emotion recognition combining acoustic features and linguistic information in a hybrid support vector machine - belief network architecture," in *Proc. ICASSP*, Montreal, QC, Canada, 2004, pp. I–577–I–580.

[8] A. Triantafyllopoulos, B. W. Schuller, G. İymen, M. Sezgin, X. He, Z. Yang, P. Tzirakis, S. Liu, S. Mertes, E. André, R. Fu, and J. Tao, "An overview of affective speech synthesis and conversion in the deep learning era," *Proceedings of the IEEE*, pp. 1–27, 2023.

[9] J. Tao and T. Tan, "Affective computing: A review," in *Proc. ACII*, Beijing, China, 2005, pp. 981–995.

[10] R. W. Picard, *Affective Computing*. MIT Press, 2000.

[11] Y. Ning, S. He, Z. Wu, C. Xing, and L. Zhang, "A review of deep learning based speech synthesis," *Applied Sciences*, vol. 9, no. 19, p. 4050, 2019.

[12] V. R. Reddy and K. S. Rao, "Better human computer interaction by enhancing the quality of text-to-speech synthesis," in *Proc. IHCI*, Kharagpur, India, 2012, pp. 1–6.

[13] P. Taylor, *Text-to-Speech Synthesis*. Cambridge University Press, 2009.

[14] I. I. Trandafilidi, T. M. Tatarnikova, and A. S. Poponin, "Speech synthesis system for people with disablities," in *Proc. WECONF*, Saint-Petersburg, Russia, 2022, pp. 1–5.

[15] H. Dudley and T. H. Tarnoczy, "The speaking machine of Wolfgang von Kempelen," *The Journal of the Acoustical Society of America*, vol. 22, no. 2, pp. 151–166, 1950.

[16] C. H. Coker, "A model of articulatory dynamics and control," *Proceedings of the IEEE*, vol. 64, no. 4, pp. 452–560, 1976.

[17] P. Seeviour, J. Holmes, and M. Judd, "Automatic generation of control signals for a parallel formant speech synthesizer," in *Proc. ICASSP*, Philadelphia, PA, USA, 1976, pp. 690–693.

[18] J. Olive, "Rule synthesis of speech from dyadic units," in *Proc. ICASSP*, Hartford, CT, USA, 1977, pp. 568–570.

[19] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in *Proc. EUROSPEECH*, Budapest, Hungary, 1999.

[20] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.

[21] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. J. Skerrv-Ryan *et al.*, "Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions," in *Proc. ICASSP*, Calgary, AB, Canada, 2018, pp. 4779–4783.

[22] X. Tan, T. Qin, F. Soong, and T. Liu, "A survey on neural speech synthesis," *arXiv preprint arXiv:2106.15561*, 2021.

[23] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2Wav: End-to-end speech synthesis," in *Proc. ICLR*, Toulon, France, 2017.

[24] R. J. Weiss, R. J. Skerry-Ryan, E. Battenberg, S. Mariooryad, and D. P. Kingma, "Wave-Tacotron: Spectrogram-free end-to-end text-to-speech synthesis," in *Proc. ICASSP*, Toronto, ON, Canada, 2021, pp. 5679–5683.

[25] R. Sproat, A. W. Black, S. Chen, S. Kumar, M. Ostendorf, and C. D. Richards, "Normalization of non-standard words," *Computer Speech & Language*, vol. 15, no. 3, pp. 287–333, 2001.

[26] R. Sproat and J. Navdeep, "RNN approaches to text normalization: A challenge," *arXiv preprint arXiv:1611.00068*, 2016.

[27] N. Xue, "Chinese word segmentation as character tagging," *Computational Linguistic & Chinese Language Processing*, vol. 8, no. 1, pp. 29–48, 2003.

[28] M. Sun and J. R. Bellegarda, "Improved POS tagging for text-to-speech synthesis," in *Proc. ICASSP*, Prague, Czech Republic, 2011, pp. 5384–5387.

[29] J. H. Jeon and Y. Liu, "Automatic prosodic events detection using syllable-based acoustic and synthetic features," in *Proc. ICASSP*, Taipei, China, 2009, pp. 4565–4568.

[30] K. Rao, F. Peng, H. Sak, and F. Beaufays, "Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks," in *Proc. ICASSP*, Brisbane, Australia, 2015, pp. 4225–4229.

[31] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. ICASSP*, Vancouver, BC, Canada, 2013, pp. 7962–7966.

[32] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep Voice 3: Scaling text-to-speech with convolutional sequence learning," in *Proc. ICLR*, Vancouver, BC, Canada, 2018.

[33] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T. Liu, "FastSpeech: Fast, robust and controllable text to speech," in *Proc. NeurIPS*, Vancouver, BC, Canada, 2019.

[34] K. Peng, W. Ping, Z. Song, and K. Zhao, "Non-autoregressive neural text-to-speech," in *Proc. ICML*, Vienna, Austria, 2020, pp. 7586–7598.

[35] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. ICASSP*, Istanbul, Turkey, 2000, pp. 1315–1318.

[36] Y. Qian, Y. Fan, W. Hu, and F. K. Soong, "On the training aspects of deep neural network (DNN) for parametric TTS synthesis," in *Proc. ICASSP*, Florence, Italy, 2014, pp. 3829–3833.

[37] Y. Fan, Y. Qian, F. Xie, and F. K. Soong, "TTS synthesis with bidirectional LSTM based recurrent neural networks," in *Proc. INTERSPEECH*, Singapore, 2014, pp. 1964–1968.

[38] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proc. ICML*, Sydney, Australia, 2017, pp. 1243–1252.

[39] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. NeurIPS*, Montreal, QC, Canada, 2014, pp. 3104–3112.

[40] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with Transformer network," in *Proc. AAAI*, Honolulu, HI, USA, 2019, pp. 6706–6713.

[41] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, San Diego, CA, USA, 2015.

[42] D. Lai and B. Lu, "Autoregressive model for time series as a deterministic dynamic system," *Predictive Analytics and Futurism*, vol. 15, pp. 7–9, 2017.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NeurIPS*, Long Beach, CA, USA, 2017.

[44] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Proc. NeurIPS*, Vancouver, BC, Canada, 2020, pp. 17 022–17 033.

[45] R. Valle, K. Shih, R. Prenger, and B. Catanzaro, "Flowtron: An autoregressive flow-based generative network for text-to-speech synthesis," *arXiv preprint arXiv:2005.05957*, 2020.

[46] J. Kim, S. Kim, J. Kong, and S. Yoon, "Glow-TTS: A generative flow for text-to-speech via monotonic alignment search," in *Proc. NeurIPS*, Vancouver, BC, Canada, 2020, pp. 8067–8077.

[47] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, and M. Kudinov, "Grad-TTS: A diffusion probabilistic model for text-to-speech," *arXiv preprint arXiv:2105.06337*, 2021.

[48] M. Jeong, H. Kim, S. J. Cheon, B. J. Choi, and N. S. Kim, "Diff-TTS: A denoising diffusion model for text-to-speech," *arXiv preprint arXiv:2104.01409*, 2021.

[49] A. de Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[50] The Linux Foundation, "TORCH: STFT," pytorch.org, https://pytorch.org/docs/stable/generated/torch.stft.html (accessed Jun. 25, 2023).

[51] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[52] H. Kawahara, "STRAIGHT, exploitation of the other aspect of VOCODER: Perceptually isomorphic decomposition of speech sounds," *Acoustical Science and Technology*, vol. 27, no. 6, pp. 349–353, 2006.

[53] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Transactions on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[54] 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," *arXiv preprint arXiv:1609.03499*, 2016.

[55] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A flow-based generative network for speech synthesis," in *Proc. ICASSP*, Brighton, UK, 2019, pp. 3617–3621.

[56] W. Ping, K. Peng, K. Zhao, and Z. Song, "WaveFlow: A compact flow-based model for raw audio," in *Proc. ICML*, Vienna, Austria, 2020, pp. 7706–7716.

[57] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," in *Proc. ICLR*, New Orleans, LA, USA, 2019.

[58] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "DiffWave: A versatile diffusion model for audio synthesis," *arXiv preprint arXiv:2009.09761*, 2020.

[59] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "WaveGrad: Estimating gradients for waveform generation," in *Proc. ICLR*, Vienna, Austria, 2021.

[60] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, "SampleRNN: An unconditional end-to-end neural audio generation model," in *Proc. ICLR*, Toulon, France, 2017.

[61] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van der Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *Proc. ICML*, Stockholm, Sweden, 2018, pp. 2410–2419.

[62] J. M. Valin and J. Skoglund, "LPCNet: Improving neural speech synthesis through linear prediction," in *Proc. ICASSP*, Brighton, UK, 2019, pp. 5891–5895.

[63] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. C. Cobo, F. Stimberg *et al.*, "Parallel WaveNet: Fast high-fidelity speech synthesis," in *Proc. ICML*, Stockholm, Sweden, 2018, pp. 3918–3926.

[64] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. NeurIPS*, Montreal, QC, Canada, 2015, pp. 577–585.

[65] B. C. J. Moore, *An Introduction to the Psychology of Hearing*. Brill, 2012.

[66] L. Deng and D. OŚhaughnessy, *Speech Processing: A Dynamic and Optimization-oriented Approach*. CRC Press, 2003.

[67] V. Popov, S. Kamenev, M. Kudinov, S. Repyevsky, T. Sadekova, V. Bushaev, V. Kryzhanovskiy, and D. Parkhomenko, "Fast and lightweight on-device TTS with Tacotron2 and LPCNet," in *Proc. INTERSPEECH*, Shanghai, China, 2020, pp. 220–224.

[68] J. Kominek and A. W. Black, "CMU ARCTIC databases for speech synthesis," Carnegie Mellon University, http://festvox.org/cmu_arctic/cmu_arctic_report.pdf (accessed Jul. 10, 2023).

[69] C. Veaux, J. Yamagishi, and K. MacDonald, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit," University of Edinburgh, https://datashare.ed.ac.uk/handle/10283/2651 (accessed Jul. 10, 2023).

[70] S. King, A. W. Black, and K. Tokuda, "The Blizzard Challenge 2011: Evaluating corpus-based speech synthesis on common databases," festvox.org, https://www.synsig.org/images/4/45/Blizzard2011_full.pdf (accessed Jul. 10, 2023).

[71] S. King, A. W. Black, K. Tokuda, and K. Prahallad, "The Blizzard Challenge 2013: Evaluating corpus-based speech synthesis on common databases," festvox.org, https://www.synsig.org/images/b/b1/Blizzard2013.pdf (accessed Jun. 27, 2023).

[72] K. Ito and L. Johnson, "The LJ Speech dataset," keithito.com, https://keithito.com/LJ-Speech-Dataset/ (accessed Jul. 11, 2023).

[73] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: An ASR corpus based on public domain audio books," in *Proc. ICASSP*, Brisbane, Australia, 2015, pp. 5206–5210.

[74] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "LibriTTS: A corpus derived from LibriSpeech for text-to-speech," in *Proc. INTERSPEECH*, Graz, Austria, 2019, pp. 1526–1530.

[75] The Centre for Speech Techonology Research, "Lessac Technologies, Inc. voice release for Blizzard 2011," University of Edinburgh, https://www.cstr.ed.ac.uk/projects/blizzard/2011/lessac_blizzard2011/ (accessed Jul. 12, 2023).

[76] M. Bernard and H. Titeux, "Phonemizer: Text to phones transcription for multiple languages in Python," *Journal of Open Source Software*, vol. 6, no. 68, p. 3958, 2021.

[77] The Carnegie Mellon University, "The cmu pronouncing dictionary," speech.cs.cmu.edu, http://www.speech.cs.cmu.edu/cgi-bin/cmudict (accessed Jul. 12, 2023).

[78] Z. Yang, X. Jing, A. Triantafyllopoulos, M. Song, I. Aslan, and B. W. Schuller, "An overview & analysis of sequence-to-sequence emotional voice conversion," in *Proc. INTERSPEECH*, Incheon, South Korea, 2022, pp. 4915–4919.

[79] R. Liu, B. Sisman, G. Gao, and H. Li, "Expressive TTS training with frame and style reconstruction loss," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1806–1818, 2021.

[80] M. Wagner and D. G. Watson, "Experimental and theoretical advances in prosody: A review," *Language and Cognitive Processes*, vol. 25, no. 7–9, pp. 905–945, 2010.

[81] R. J. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. J. Weiss, R. Clark, and R. A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," in *Proc. ICML*, Stockholm, Sweden, 2018, pp. 7471–7480.

[82] Y. Wang, D. Stanton, Y. Zhang, R. J. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *Proc. ICML*, Stockholm, Sweden, 2018, pp. 5180–5189.

[83] J. E. Cahn, "The generation of affect in synthesized speech," *Journal of the American Voice I/O Society*, vol. 8, no. 1, 1990.

[84] N. Campbell, "Expressive/affective speech synthesis," *Springer Handbook of Speech Processing*, pp. 505–518, 2008.

[85] I. R. Murray, "Simulating emotion in synthetic speech," *Ph.D. Dissertation*, 1989.

[86] J. E. Cahn, "Generating expression in synthesized speech," *Ph.D. Dissertation*, 1989.

[87] F. Burkhardt and W. F. Sendlmeier, "Verification of acoustical correlates of emotional speech using formant-synthesis," in *Proc. ITRW*, Newcastle, UK, 2000.

[88] A. W. Black, "Unit selection and emotional speech," in *Proc. INTERSPEECH*, Geneva, Switzerland, 2003.

[89] A. Iida, N. Campbell, F. Higuchi, and M. Yasumura, "A corpus-based speech synthesis system with emotion," *Speech Communication*, vol. 40, no. 1-2, pp. 161–187, 2003.

[90] M. Tachibana, J. Yamagishi, K. Onishi, T. Masuko, and T. Kobayashi, "HMM-based speech synthesis with various speaking styles using model interpolation," in *Proc. International Conference on Speech Prosody*, Nara, Japan, 2004.

[91] S. An, Z. Ling, and L. Dai, "Emotional statistical parametric speech synthesis using LSTM-RNNs," in *Proc. APSIPA-ASC*, Kuala Lumpur, Malaysia, 2017, pp. 1613–1616.

[92] J. Lorenzo-Trueba, G. E. Henter, S. Takaki, J. Yamagishi, Y. Morino, and Y. Ochiai, "Investigating different representations for modeling and controlling multiple emotions in DNN-based speech synthesis," *Speech Communication*, vol. 99, pp. 135–143, 2018.

[93] O. Kwon, I. Jang, C. Ahn, and H. Kang, "An effective style token weight control technique for end-to-end emotional speech synthesis," *IEEE Signal Processing Letters*, vol. 26, no. 9, pp. 1383–1387, 2019.

[94] ——, "Emotional speech synthesis based on style embedded Tacotron2 framework," in *Proc. ITC-CSCC*, Jeju Island, South Korea, 2019, pp. 1–4.

[95] H. Choi, S. Park, J. Park, and M. Hahn, "Multi-speaker emotional acoustic modeling for CNN-based speech synthesis," in *Proc. ICASSP*, Brighton, UK, 2019, pp. 6950–6954.

[96] L. Chen and A. Rudnicky, "Fine-grained style control in Transformer-based text-to-speech synthesis," in *Proc. ICASSP*, Singapore, 2022, pp. 7907–7911.

[97] Y. Lee, A. Rabiee, and S. Lee, "Emotional end-to-end neural speech synthesizer," in *Proc. NeurIPS*, Long Beach, CA, USA, 2017.

[98] N. Tits, K. E. Haddad, and T. Dutoit, "Exploring transfer learning for low resource emotional TTS," in *Proc. IntelliSys*, London, UK, 2019, pp. 52–60.

[99] K. Zhou, B. Sisman, R. Liu, and H. Li, "Emotional voice conversion: Theory, databases and ESD," *Speech Communication*, vol. 137, pp. 1–18, 2022.

[100] A. Adigwe, N. Tits, K. E. Haddad, S. Ostadabbas, and T. Dutoit, "The emotional voice database: Towards controlling the emotion dimension in voice generation systems," *arXiv preprint arXiv:1806.09514*, 2018.

[101] R. F. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Proc. PACRIM*, Victoria, BC, Canada, 1993, pp. 125–128.

[102] T. Nakatani, S. Amano, T. Irino, K. Ishizuka, and T. Kondo, "A method for fundamental frequency estimation and voicing decision: Application to infant utterances recorded in real acoustical environments," *Speech Communication*, vol. 50, no. 3, pp. 203–214, 2008.

[103] C. Busso, M. Bulut, C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. Chang, S. Lee, and S. Narayanan, "IEMOCAP: Interactive emotional dyadic motion capture database," *Journal of Language Resources and Evaluation*, vol. 42, no. 4, pp. 335–359, 2008.

[104] C. Busso, S. Parthasarathy, A. Burmania, M. Wahab, N. Sadoughi, and E. Provost, "MSP-IMPROV: An acted corpus of dyadic interactions to study emotion perception," *IEEE Transactions on Affective Computing*, vol. 8, no. 1, pp. 119–130, 2017.

[105] S. R. Livingstone and F. A. Russo, "The ryerson audio-visual database of emotional speech and song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in north american english," *PLOS One*, vol. 13, no. 5, 2018.

[106] H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma, "CREMA-D: Crowd-sourced emotional multimodal actors dataset," *IEEE Transactions on Affective Computing*, vol. 5, no. 4, pp. 377–390, 2014.

[107] F. Ringeval, A. Sonderegger, J. Sauer, and D. Lalanne, "Introducing the RECOLA multimodal corpus of remote collaborative and affective interactions," in *Proc. FG*, Shanghai, China, 2013, pp. 1–8.

[108] E. Parada-Cabaleiro, G. Costantini, A. Batliner, M. Schmitt, and B. Schuller, "DEMoS: An italian emotional speech corpus," *Language Resources and Evaluation*, pp. 1–43, 2019.

[109] F. Burkhardt, A. Paeschke, M. Rolfes, W. Sendlmeier, and B. Weiss, "A database of German emotional speech," in *Proc. INTERSPEECH*, Lisbon, Portugal, 2005, pp. 1517–1520.

[110] S. Steidl, "Automatic classification of emotion related user states in spontaneous children's speech," 2009.

[111] H. Gunes, B. W. Schuller, M. Pantic, and R. Cowie, "Emotion representation, analysis and synthesis in continuous space: A survey," in *Proc. FG*, Santa Barbara, CA, USA, 2011, pp. 827–834.

[112] J. A. Russell, "A circumplex model of affect," *Journal of Personality and Social Psychology*, vol. 39, no. 6, pp. 1161–1178, 1980.

[113] A. Mehrabian, "Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament," *Current Psychology*, vol. 14, pp. 261–292, 1996.

[114] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[115] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. D. Laroussihe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficiency transfer learning for NLP," in *Proc. ICML*, Long Beach, CA, USA, 2019, pp. 2790–2799.

[116] Z. Alyafeai, M. S. AlShaibani, and I. Ahmad, "A survey on transfer learning in natural language processing," *arXiv preprint arXiv:2007.04239*, 2020.

[117] A. Brodzicki, M. Piekarski, D. Kucharski, J. Jaworek-Korjakowska, and M. Gorgon, "Transfer learning methods as a new approach in computer vision tasks with small datasets," *Foundations of Computing and Decision Sciences*, vol. 45, no. 3, pp. 179–193, 2020.

[118] S. Kentsch, M. L. L. Caceres, D. Serrano, F. Roure, and Y. Diez, "Computer vision and deep learning techniques for the analysis of drone-acquired forest images, a transfer learning study," *Remote Sensing*, vol. 12, no. 8, 2020.

[119] J. Kunze, L. Kirsch, I. Kurenkov, A. Krug, J. Johannsmeier, and S. Stober, "Transfer learning for speech recognition on a budget," *arXiv preprint arXiv:1706.00290*, 2017.

[120] H. Inaguma, J. Cho, M. K. Baskar, T. Kawahara, and S. Watanabe, "Transfer learning of language-independent end-to-end ASR with language model fusion," in *Proc. ICASSP*, Brighton, UK, 2019, pp. 6096–6100.

[121] Y. Jia, Y. Zhang, R. J. Weiss, Q. Wang, J. Shen, F. Ren, Z. Chen, P. Nguyen, R. Pang, I. L. Moreno, and Y. Wu, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Proc. NeurIPS*, Montreal, QC, Canada, 2018.

[122] Y. Wang, R. Yi, Y. Tai, C. Wang, and L. Ma, "CtlGAN: Few-shot artistic portraits generation with contrastive transfer learning," *arXiv preprint arXiv:2203.08612*, 2022.

[123] J. Deng, Z. Zhang, E. Marchi, and B. W. Schuller, "Sparse autoencoder-based feature transfer learning for speech emotion recognition," in *Proc. ACII*, Geneva, Switzerland, 2013, pp. 511–156.

[124] S. Latif, R. Rana, S. Younis, J. Qadir, and J. Epps, "Transfer learning for improving speech emotion classification accuracy," *arXiv preprint arXiv:1801.06353*, 2018.

[125] M. Gerczuk, S. Amiriparian, S. Ottl, and B. W. Schuller, "EmoNet: A transfer learning framework for multi-corpus speech emotion recognition," *IEEE Transactions on Affective Computing*, vol. 14, no. 2, pp. 1472–1487, 2023.

[126] H. Zhang and Y. Lin, "Unsupervised learning for sequence-to-sequence text-to-speech for low-resource languages," *arXiv preprint arXiv:2008.04549*, 2020.

[127] C. Miao, S. Liang, M. Chen, J. Ma, S. Wang, and J. Xiao, "Flow-TTS: A non-autoregressive network for text to speech based on flow," in *Proc. ICASSP*, Barcelona, Spain, 2020, pp. 7209–7213.

[128] X. Jing, Y. Chang, Z. Yang, A. Triantafyllopoulos, and B. W. Schuller, "U-DiT TTS: U-diffusion vision transformer for text-to-speech," *arXiv preprint arXiv:2305.13195*, 2023.

[129] Y. Saito, Y. Ijima, K. Nishida, and S. Takamichi, "Non-parallel voice conversion using variational autoencoders conditioned by phonetic posteriorgrams and d-vectors," in *Proc. ICASSP*, Calgary, AB, Canada, 2018, pp. 5274–5278.

[130] B. Sisman, J. Yamagishi, S. King, and H. Li, "An overview of voice conversion and its challenges: From statistical modeling to deep learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 132–157, 2021.

[131] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 2, pp. 131–142, 1998.

[132] H. Valbret, E. Moulines, and J. Tubach, "Voice transformation using PSOLA technique," *Speech Communication*, vol. 11, no. 2–3, pp. 175–187, 1992.

[133] D. Sundermann, H. Ney, and H. Hoge, "VTLN-based cross-language voice conversion," in *Proc. ASRU*, Saint Thomas, VI, USA, 2003, pp. 676–681.

[134] B. Sisman, M. Zhang, and H. Li, "Group sparse representation with WaveNet vocoder adaptation for spectrum and prosody conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 6, pp. 1085–1097, 2019.

[135] A. Mouchtaris, J. V. der Spiegel, and P. Mueller, "Nonparallel training for voice conversion based on a parameter adaptation approach," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 3, pp. 952–963, 2006.

[136] T. Kinnunen, L. Juvela, P. Alku, and J. Yamagishi, "Non-parallel voice conversion using i-vector PLDA: Towards unifying speaker verification and transformation," in *Proc. ICASSP*, New Orleans, LA, USA, 2017, pp. 5535–5539.

[137] L. Chen, Z. Ling, L. Liu, and L. Dai, "Voice conversion using deep neural networks with layer-wise generative training," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1859–1872, 2014.

[138] L. Sun, S. Kang, K. Li, and H. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," in *Proc. ICASSP*, Brisbane, Australia, 2015, pp. 4869–4873.

[139] H. Kameoka, K. Tanaka, D. Kwaśny, T. Kaneko, and N. Hojo, "ConvS2S-VC: Fully convolutional sequence-to-sequence voice conversion," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1849–1863, 2020.

[140] T. Kaneko and H. Kameoka, "CycleGAN-VC: Non-parallel voice conversion using cycle-consistent adversarial networks," in *Proc. EUSIPCO*, Rome, Italy, 2018, pp. 2100–2104.

[141] M. Zhang, X. Wang, F. Fang, H. Li, and J. Yamagishi, "Joint training framework for text-to-speech and voice conversion using multi-source Tacotron and WaveNet," in *Proc. INTERSPEECH*, Graz, Austria, 2019, pp. 1298–1302.

[142] H. Miyoshi, Y. Saito, S. Takamichi, and H. Saruwatari, "Voice conversion using sequence-to-sequence learning of context posterior probabilities," in *Proc. INTERSPEECH*, Stockholm, Sweden, 2017, pp. 1268–1272.

[143] C. Hsu, H. Hwang, Y. Wu, Y. Tsao, and H. Wang, "Voice conversion from non-parallel corpora using variational auto-encoder," in *Proc. APSIPA-ASC*, Jeju Island, South Korea, 2016, pp. 1–6.

[144] ——, "Voice conversion from unaligned corpora using variational autoencoding Wasserstein generative adversarial networks," in *Proc. INTERSPEECH*, Stockholm, Sweden, 2017, pp. 3364–3368.

[145] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[146] Y. Cao, Z. Liu, M. Chen, J. Ma, S. Wang, and J. Xiao, "Nonparallel emotional speech conversion using VAE-GAN," in *Proc. INTERSPEECH*, Shanghai, China, 2020, pp. 3406–3410.

[147] S. Liu, Y. Cao, and H. Meng, "Emotional voice conversion with cycle-consistent adversarial network," *arXiv preprint arXiv:2004.03781*, 2020.

[148] P. Ekman, "An argument for basic emotions," *Cognition & Emotion*, vol. 6, no. 3–4, pp. 169–200, 1992.

[149] E. Brunswik, "Historical and thematic relations of psychology to other sciences," *The Scientific Monthly*, vol. 83, no. 3, pp. 169–200, 1956.

[150] Y. Xu, "Speech prosody: A methodological review," *Journal of Speech Sciences*, vol. 1, no. 1, pp. 85–115, 2011.

[151] K. Zhou, B. Sisman, M. Zhang, and H. Li, "Converting anyone's emotion: Towards speaker-independent emotional voice conversion," in *Proc. INTERSPEECH*, Shanghai, China, 2020, pp. 3416–3420.

[152] J. Teutenberg, C. Watson, and P. Riddle, "Modelling and synthesising F0 contours with the discrete cosine transform," in *Proc. ICASSP*, Las Vegas, NV, USA, 2008, pp. 3973–3976.

[153] N. Obin and J. Beliao, "Sparse coding of pitch contours with deep auto-encoders," in *Proc. International Conference on Speech Prosody*, Poznan, Poland, 2018, pp. 799–803.

[154] Z. Luo, T. Takiguchi, and Y. Ariki, "Emotional voice conversion using deep neural networks with MCC and F0 features," in *Proc. ICIS*, Dublin, Ireland, 2016, pp. 1–5.

[155] J. Gao, D. Chakraborty, H. Tembine, and O. Olaleye, "Nonparallel emotional speech conversion," in *Proc. INTERSPEECH*, Graz, Austria, 2019, pp. 2858–2862.

[156] K. Zhou, B. Sisman, and H. Li, "Transforming spectrum and prosody for emotional voice conversion with non-parallel training data," *arXiv preprint arXiv:2002.00198*, 2020.

[157] Y. Wang, W. Song, W. Tao, A. Liotta, D. Yang, X. Li, S. Gao, Y. Sun, W. Ge, W. Zhang, and W. Zhang, "A systematic review on affective computing: Emotion models, databases, and recent advances," *Information Fusion*, vol. 83–84, pp. 19–52, 2022.

[158] H. Ming, D. Huang, M. Dong, H. Li, L. Xie, and S. Zhang, "Fundamental frequency modeling using wavelets for emotional voice conversion," in *Proc. ACII*, Xi'an, China, 2015, pp. 804–809.

[159] H. Ming, D. Huang, L. Xie, J. Wu, M. Dong, and H. Li, "Deep bidirectional LSTM modeling of timbre and prosody for emotional voice conversion," in *Proc. INTERSPEECH*, San Francisco, CA, USA, 2016, pp. 2453–2457.

[160] Z. Luo, J. Chen, T. Takiguchi, and Y. Ariki, "Emotional voice conversion with adaptive scales F0 based on wavelet transform using limited amount of emotional data," in *Proc. INTERSPEECH*, Stockholm, Sweden, 2017, pp. 3399–3403.

[161] ——, "Emotional voice conversion using dual supervised adversarial networks with continuous wavelet transform F0 features," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 10, pp. 1535–1548, 2019.

[162] G. Rizos, A. Baird, M. Elliott, and B. W. Schuller, "StarGAN for emotional speech conversion: Validated by data augmentation of end-to-end emotion recognition," in *Proc. ICASSP*, Barcelona, Spain, 2020, pp. 3502–3506.

[163] Z. Luo, J. Chen, T. Takiguchi, and Y. Ariki, "Emotional voice conversion using neural networks with arbitrary scales F0 based on wavelet transform," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2017, pp. 1–13, 2017.

[164] K. Zhou, B. Sisman, R. Liu, and H. Li, "Seen and unseen emotional style transfer for voice conversion with a new emotional speech dataset," in *Proc. ICASSP*, Toronto, ON, Canada, 2021, pp. 920–924.

[165] Z. Luo, J. Chen, T. Nakashika, T. Takiguchi, and Y. Ariki, "Emotional voice conversion using neural networks with different temporal scales of F0 based on wavelet transform," in *Proc. ISCA SSW*, Sunnyvale, CA, USA, 2016, pp. 153–158.

[166] L. Sun, K. Li, H. Wang, S. Kang, and H. Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," in *Proc. ICME*, Seattle, WA, USA, 2016, pp. 1–6.

[167] H. Lu, Z. Wu, R. Li, S. Kang, J. Jia, and H. Meng, "A compact framework for voice conversion using WaveNet conditioned on phonetic posteriorgrams," in *Proc. ICASSP*, Brighton, UK, 2019, pp. 6810–6814.

[168] librosa development team, "librosa.feature.mfcc – librosa 0.10.1 documentation," librosa.org, https://librosa.org/doc/main/generated/librosa.feature.mfcc.html (accessed Mar. 1, 2023).

[169] The Linux Foundation, "MFCC – Torchaudio 2.2.0.dev20240301 documentation," pytorch.org, https://pytorch.org/audio/main/generated/torchaudio.transforms.MFCC.html (accessed Jun. 25, 2023).

[170] M. Morise, G. Miyashita, and K. Ozawa, "Low-dimensional representation of spectral envelope without deterioration for full-band speech analysis/synthesis system," in *Proc. INTERSPEECH*, Stockholm, Sweden, 2017, pp. 409–413.

[171] T. Yoshimura, T. Fujimoto, K. Oura, and K. Tokuda, "SPTK4: An open-source software toolkit for speech signal processing," in *Proc. ISCA SSW*, Grenoble, France, 2023, pp. 211–217.

[172] B. W. Schuller, B. Vlasenko, F. Eyben, G. Rigoll, and A. Wendemuth, "Acoustic emotion recognition: A benchmark comparison of performances," in *Proc. IEEE Workshop on Automatic Speech Recognition Understanding*, Pattaya, Thailand, 2009, pp. 552–557.

[173] B. W. Schuller and A. Batliner, *Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing*. John Wiley & Sons, 2013.

[174] J. Deng, "Feature transfer learning for speech emotion recognition," Ph.D. dissertation, Technische Universität München, 2016.

[175] F. Eyben, M. Wöllmer, and B. W. Schuller, "OpenSMILE: The Munich versatile and fast open-source audio feature extractor," in *Proc. ACM MM*, Florence, Italy, 2010, pp. 1459–1462.

[176] M. Elgaar, J. Park, and S. W. Lee, "Multi-speaker and multi-domain emotional voice conversion using factorized hierarchical variational autoencoder," in *Proc. ICASSP*, Barcelona, Spain, 2020, pp. 7769–7773.

[177] Z. Yang, K. Qian, Z. Ren, A. Baird, Z. Zhang, and B. W. Schuller, "Learning multi-resolution representations for acoustic scene classification via neural networks," in *Proc. CSMT*, Harbin, China, 2019, pp. 133–143.

[178] Z. Ren, V. Pandit, K. Qian, Z. Yang, Z. Zhang, and B. W. Schuller, "Deep sequential image features on acoustic scene classification," in *Proc. DCASE*, Munich, Germany, 2017, pp. 113–117.

[179] K. Qian, Z. Ren, V. Pandit, Z. Yang, Z. Zhang, and B. W. Schuller, "Wavelets revisited for the classification of acoustic scenes," in *Proc. DCASE*, Munich, Germany, 2017, pp. 108–112.

[180] K. Qian, C. Janott, V. Pandit, Z. Zhang, C. Heiser, W. Hohenhorst, M. Herzog, W. Hemmert, and B. W. Schuller, "Classification of the excitation location of snore sounds in the upper airway by acoustic multi-feature analysis," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 8, pp. 1731–1741, 2017.

[181] R. Aihara, R. Takashima, T. Takiguchi, and Y. Ariki, "GMM-based emotional voice conversion using spectrum and prosody features," *American Journal of Signal Processing*, vol. 2, no. 5, pp. 134–138, 2012.

[182] J. Tao, Y. Kang, and A. Li, "Prosody conversion from neutral speech to emotional speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1145–1154, 2006.

[183] R. Aihara, R. Ueda, T. Takiguchi, and Y. Ariki, "Exemplar-based emotional voice conversion using non-negative matrix factorization," in *Proc. APSIPA-ASC*, Angkor Wat, Cambodia, 2014, pp. 1–7.

[184] M. Müller, "Dynamic time warping," *Information Retrieval for Music and Motion*, pp. 69–84, 2007.

[185] H. Ye and S. Young, "Voice conversion for unknown speakers," in *Proc. ICSLP*, Jeju Island, South Korea, 2004.

[186] C. Robinson, N. Obin, and A. Roebel, "Sequence-to-sequence modelling of F0 for speech emotion conversion," in *Proc. ICASSP*, Brighton, UK, 2019, pp. 6830–6834.

[187] T. Kim, S. Cho, S. Choi, S. Park, and S. Lee, "Emotional voice conversion using multitask learning with text-to-speech," in *Proc. ICASSP*, Barcelona, Spain, 2020, pp. 7774–7778.

[188] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. ICASSP*, Vancouver, BC, Canada, 2013, pp. 6645–6649.

[189] M. Wöllmer, M. Kaiser, F. Eyben, B. W. Schuller, and G. Rigoll, "LSTM-modeling of continuous emotions in an audiovisual affect recognition framework," *Image and Vision Computing*, vol. 31, no. 2, pp. 153–163, 2013.

[190] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM–a tutorial into long short-term memory recurrent neural networks," *arXiv preprint arXiv:1909.09586*, 2019.

[191] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. ICML*, Atlanta, GA, USA, 2013, pp. 1310–1318.

[192] A. Graves, N. Jaitly, and A. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Proc. ASRU*, Olomouc, Czech Republic, 2013, pp. 273–278.

[193] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[194] V. Dissanayake, H. Zhang, M. Billinghurst, and S. Nanayakkara, "Speech emotion recognition 'in the wild' using an autoencoder," in *Proc. INTERSPEECH*, Shanghai, China, 2020, pp. 526–530.

[195] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[196] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communication of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[197] ——, "Generative adversarial nets," in *Proc. NeurIPS*, Montreal, QC, Canada, 2014.

[198] L. Wang, W. Chen, W. Yang, F. Bi, and F. R. Yu, "A state-of-the-art review on image synthesis with generative adversarial networks," *IEEE Access*, vol. 8, pp. 63 514–63 537, 2020.

[199] A. Wali, Z. Alamgir, S. Karim, A. Fawaz, M. B. Ali, M. Adan, and M. Mujtaba, "Generative adversarial networks for speech processing: A review," *Computer Speech & Language*, vol. 72, p. 101308, 2022.

[200] A. Torfi, R. A. Shirvani, Y. Keneshloo, N. Tavaf, and E. A. Fox, "Natural language processing advancements by deep learning: A survey," *arXiv preprint arXiv:2003.01200*, 2020.

[201] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. ICCV*, Venice, Italy, 2017, pp. 2223–2232.

[202] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "CycleGAN-VC2: Improved CycleGAN-based non-parallel voice conversion," in *Proc. ICASSP*, Brighton, UK, 2019, pp. 6820–6824.

[203] M. Morise, "D4C, a band-aperiodicity estimator for high-quality speech synthesis," *Speech Communication*, vol. 84, pp. 57–65, 2016.

[204] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. ICML*, Sydney, Australia, 2017, pp. 933–941.

[205] J. Duda, "Gaussian autoencoder," *arXiv preprint arXiv:1811.04751*, 2018.

[206] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[207] A. Asperti and M. Trentin, "Balancing reconstruction error and Kullback-Leibler divergence in variational autoencoders," *IEEE Access*, vol. 8, pp. 199 440–199 448, 2020.

[208] J. Lu, K. Zhou, B. Sisman, and H. Li, "VAW-GAN for singing voice conversion with non-parallel training data," in *Proc. APSIPA-ASC*, Auckland, New Zealand, 2020, pp. 514–519.

[209] E. A. Albadawy and S. Lyu, "Voice conversion using speech-to-speech neuro-style transfer," in *Proc. INTERSPEECH*, Shanghai, China, 2020, pp. 4726–4730.

[210] M. Akbari and J. Liang, "Semi-recurrent CNN-based VAE-GAN for sequential data generation," in *Proc. ICASSP*, Calgary, AB, Canada, 2018, pp. 2321–2325.

[211] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, "CVAE-GAN: Fine-grained image generation through asymmetric training," in *Proc. ICCV*, Venice, Italy, 2017, pp. 2745–2754.

[212] T. J. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *Proc. ASRU*, Merano, Italy, 2009, pp. 421–426.

[213] G. Zhao, S. Sonsatt, J. Levis, E. Chukharev-Hudilainen, and R. Gutierrez-Osuna, "Accent conversion using phonetic posteriorgrams," in *Proc. ICASSP*, Calgary, AB, Canada, 2018, pp. 5314–5318.

[214] G. Zhao and R. Gutierrez-Osuna, "Using phonetic posteriorgram based frame pairing for segmental accent conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 10, pp. 1649–1660, 2019.

[215] H. Guo, H. Lu, N. Hu, C. Zhang, S. Yang, L. Xie, D. Su, and D. Yu, "Phonetic posteriorgrams based many-to-many singing voice conversion via adversarial training," *arXiv preprint arXiv:2012.01837*, 2020.

[216] S. Liu, Y. Cao, D. Su, and H. Meng, "DiffSVC: A diffusion probabilistic model for singing voice conversion," in *Proc. ASRU*, Cartagena, Colombia, 2021, pp. 741–748.

[217] A. Pettarin, "aeneas," readbeyond.it, https://www.readbeyond.it/aeneas/ (accessed Aug. 27, 2023).

[218] C. Liu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. ICASSP*, Calgary, AB, Canada, 2018, pp. 4774–4778.

[219] R. Liu, X. Chen, and X. Wen, "Voice conversion with Transformer network," in *Proc. ICASSP*, Barcelona, Spain, 2020, pp. 7759–7763.

[220] W. Huang, T. Hayashi, Y. Wu, H. Kameoka, and T. Toda, "Voice transformer network: Sequence-to-sequence voice conversion using Transformer with text-to-speech pretraining," in *Proc. INTERSPEECH*, Shanghai, China, 2020, pp. 4676–4680.

[221] Z. Zhao, J. Liang, Z. Zheng, L. Yan, Z. Yang, W. Ding, and D. Huang, "Improving model stability and training efficiency in fast, high quality expressive voice conversion system," in *Proc. ICMI*, Montreal, QC, Canada, 2021, pp. 75–79.

[222] H. Choi and M. Hahn, "Sequence-to-sequence emotional voice conversion with strength control," *IEEE Access*, vol. 9, pp. 42 674–42 687, 2021.

[223] K. Zhou, B. Sisman, and H. Li, "Limited data emotional voice conversion leveraging text-to-speech: Two-stage sequence-to-sequence training," in *Proc. INTERSPEECH*, Brno, Czechia, 2021, pp. 811–815.

[224] K. Zhou, B. Sisman, R. Rana, B. W. Schuller, and H. Li, "Emotion intensity and its control for emotional voice conversion," *IEEE Transactions on Affective Computing*, vol. 14, no. 1, pp. 31–48, 2022.

[225] J. Zhang, Z. Ling, Y. Jing, L. Liu, C. Liang, and L. Dai, "Improving sequence-to-sequence voice conversion by adding text-supervision," in *Proc. ICASSP*, Brighton, UK, 2019, pp. 6785–6789.

[226] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *Proc. ICASSP*, New Orleans, LA, USA, 2017, pp. 4835–4839.

[227] C. Veaux and X. Rodet, "Intonation conversion from neutral to expressive speech," in *Proc. INTERSPEECH*, Florence, Italy, 2011, pp. 2765–2768.

[228] R. Yamamoto, E. Song, and J. Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *Proc. ICASSP*, Barcelona, Spain, 2020, pp. 6199–6203.

[229] The Analysis/Synthesis Team, "Super VP," Institute for Research and Coordination in Acoustics/Music, http://anasynth.ircam.fr/home/english/software/supervp (accessed Mar. 24, 2023).

[230] E. Wittenberg, M. Bharel, J. F. P. Bridges, Z. Ward, and L. Weinreb, "Using best-worst scaling to understand patient priorities: A case example of papanicolaou tests for homeless women," *The Annals of Family Medicine*, vol. 14, no. 4, pp. 359–364, 2016.

[231] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, "Independently recurrent neural network (IndRNN): Building a longer and deeper RNN," in *Proc. CVPR*, Salt Lake City, UT, USA, 2018, pp. 5457–5466.

[232] Q. Wang and A. B. Chan, "CNN+CNN : Convolutional decoders for image captioning," *arXiv preprint arXiv:1805.09019*, 2018.

[233] X. He, Y. Chen, and Z. Lin, "Spatial-spectral Transformer for hyperspectral image classification," *Remote Sensing*, vol. 13, no. 3, p. 498, 2021.

[234] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Las Vegas, NV, USA, 2016, pp. 770–778.

[235] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text Transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[236] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proc. ACL*, Florence, Italy, 2019, pp. 2978–2988.

[237] A. Roy, M. Saffar, A. Vaswani, and D. Grangier, "Efficient content-based sparse attention with routing Transformers," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 53–68, 2021.

[238] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. ICLR*, Online, 2021.

[239] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "ViViT: A video vision Transformer," in *Proc. ICCV*, Online, 2021, pp. 6836–6846.

[240] A. Gulati, J. Qin, C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented Transformer for speech recognition," in *Proc. INTERSPEECH*, Shanghai, China, 2020, pp. 5036–5040.

[241] L. H. Li, M. Yatskar, D. Yin, C. Hsieh, and K. Chang, "VisualBERT: A simple and performant baseline for vision and language," *arXiv preprint arXiv:1908.03557*, 2019.

[242] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *Proc. ICML*, Online, 2021, pp. 8821–8831.

[243] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional Transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2019.

[244] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, and et al., "GPT-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[245] X. Yuan, T. Wang, C. Gulcehre, A. Sordoni, P. Bachman, S. Zhang, S. Subramanian, and A. Trischler, "Machine comprehension by text-to-text neural question generation," in *Proc. RepL4NLP*, Vancouver, BC, Canada, 2017, pp. 15–25.

[246] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[247] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[248] P. M. Sørensen, B. Epp, and T. May, "A depthwise separable convolutional neural network for keyword spotting on an embedded system," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 1, no. 10, p. 10, 2020.

[249] M. Song, Z. Yang, A. Baird, E. Parada-Cabaleiro, Z. Zhang, Z. Zhao, and B. W. Schuller, "Audiovisual analysis for recognising frustration during game-play: Introducing the multimodal game frustration database," in *Proc. ACII*, Cambridge, UK, 2019, pp. 517–523.

[250] A. Pandey, C. Liu, Y. Wang, and Y. Saraf, "Dual application of speech enhancement for automatic speech recognition," in *Proc. SLT*, Shenzhen, China, 2021, pp. 223–228.

[251] D. S. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. INTERSPEECH*, Graz, Austria, 2019, pp. 2613–2617.

[252] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Proc. NeurIPS*, Montreal, QC, Canada, 2015, pp. 1171–1179.

[253] F. Kreuk, A. Polyak, J. Copet, E. Kharitonov, T. A. Nguyen, M. Rivière, W. Hsu, A. Mohamed, E. Dupoux, and Y. Adi, "Textless speech emotion conversion using discrete & decomposed representations," in *Proc. EMNLP*, Abu Dhabi, UAE, 2022, pp. 11 200–11 214.

[254] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *Proc. ICML*, Online, 2021, pp. 12 310–12 320.

[255] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 11, pp. 2579–2605, 2008.

[256] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li, L. He, S. Zhao, and F. Wei, "Neural codec language models are zero-shot text to speech synthesizers," *arXiv preprint arXiv:2301.02111*, 2023.

[257] W. C. Huang, E. Cooper, Y. Tsao, H. Wang, T. Toda, and J. Yamagishi, "The VoiceMOS Challenge 2022," in *Proc. INTERSPEECH*, Incheon, South Korea, 2022, pp. 4536–4540.

[258] A. Metallinou, M. Wöllmer, A. Katsamanis, F. Eyben, B. W. Schuller, and S. Narayanan, "Context-sensitive learning for enhanced audiovisual emotion classification," *IEEE Transactions on Affective Computing*, vol. 3, no. 2, pp. 184–198, 2012.

[259] G. Smith, *Essential Statistics, Regression, and Econometrics*. Academic Press, 2012.

[260] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Proc. NeurIPS*, Vancouver, BC, Canada, 2020.

[261] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T. Liu, "FastSpeech 2: Fast and high-quality end-to-end text to speech," in *Proc. ICLR*, Vienna, Austria, 2021.

[262] C. Cui, Y. Ren, J. Liu, F. Chen, R. Huang, M. Lei, and Z. Zhao, "EMOVIE: A Mandarin emotion speech dataset with a simple emotional text-to-speech model," in *Proc. INTERSPEECH*, Brno, Czechia, 2021, pp. 2766–2770.

[263] D. Diatlova and V. Shutov, "EmoSpeech: Guiding FastSpeech2 towards emotional text to speech," in *Proc. ISCA SSW*, Grenoble, France, 2023, pp. 106–112.

[264] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *Proc. ICML*, Lille, France, 2015, pp. 1530–1538.

[265] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, "Improved variational inference with inverse autoregressive flow," in *Proc. NeurIPS*, Barcelona, Spain, 2016.

[266] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1×1 convolutions," in *Proc. NeurIPS*, Montreal, QC, Canada, 2018.

[267] S. Kim, S. Lee, J. Song, J. Kim, and S. Yoon, "FloWaveNet: A generative flow for raw audio," in *Proc. ICML*, Long Beach, CA, USA, 2019, pp. 3370–3378.

[268] H. Cao, C. Tan, Z. Gao, Y. Xu, G. Chen, P. Heng, and S. Z. Li, "A survey on generative diffusion models," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 7, pp. 2814–2830, 2024.

[269] F. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, "Diffusion models in vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 9, pp. 10 850–10 869, 2023.

[270] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. NeurIPS*, Online, 2020.

[271] H. Kim, S. Kim, and S. Yoon, "Guided-TTS: A diffusion model for text-to-speech via classifier guidance," in *Proc. ICML*, Baltimore, MD, USA, 2022, pp. 11 119–11 133.

[272] S. Kim, H. Kim, and S. Yoon, "Guided-TTS 2: A diffusion model for high-quality adaptive text-to-speech with untranscribed data," *arXiv preprint arXiv:2205.15370*, 2022.

[273] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," openai.com, https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf (accessed Apr. 10, 2023).

[274] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," openai.com, https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf (accessed Apr. 10, 2023).

[275] OpenAI, "ChatGPT can now see, hear, and speak," openai.com, https://openai.com/index/chatgpt-can-now-see-hear-and-speak/ (accessed Apr. 10, 2024).

[276] W. Hsu, B. Bolte, Y. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[277] Y. Chung, Y. Zhang, W. Han, C. Chiu, J. Qin, R. Pang, and Y. Wu, "w2v-BERT: Combining contrastive learning and masked language modeling for self-supervised speech pre-training," in *Proc. ASRU*, Cartagena, Colombia, 2021, pp. 244–250.

[278] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi, and N. Zeghidour, "AudioLM: A language modeling approach to audio generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2523–2533, 2023.

[279] Z. Zhang, L. Zhou, C. Wang, S. Chen, Y. Wu, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li, L. He, S. Zhao, and F. Wei, "Speak foreign languages with your own voice: Cross-lingual neural codec language modeling," *arXiv preprint arXiv:2303.03926*, 2023.

[280] Z. Hu, L. Wang, Y. Lan, W. Xu, E. Lim, L. Bing, X. Xu, S. Poria, and R. K. Lee, "LLM-adapters: An adapter family for parameter-efficient fine-tuning of large language models," in *Proc. EMNLP*, Singapore, 2023, pp. 5254–5276.

[281] M. Gerczuk, A. Triantafyllopoulos, S. Amiriparian, A. Kathan, J. Bauer, M. Berking, and B. W. Schuller, "Zero-shot personalization of speech foundation models for depressed mood monitoring," *Patterns*, vol. 4, no. 11, p. 100873, 2023.