國立成功大學工學院

工程科學系

碩士論文

一個調整文字轉語音模型所產生之語音語速之系統

A System for Modifying the Duration of Synthesized Speech from Text-To-Speech Models.

研 究 生：楊崇文

指導教授：賴槿峰 博士

中華民國 一一二年 七月

國立成功大學

碩士論文

一個調整文字轉語音模型所產生之語音語速之系統

A System for Modifying the Duration of Synthesized Speech from Text-To-Speech Models

研究生：楊崇文

本論文業經審查及口試合格特此證明

論文考試委員：

賴槿峰　蘇豪生

賴盈動　陳世輝　李家瑋

指導教授：賴槿峰

單位主管：潘文峰

（單位主管是否簽章授權由各院、系（所、學位學程）自訂）

中 華 民 國 112 年 7 月 26 日

摘要

在過去幾年裡，文字轉語音因為其多樣的應用性而受到許多的研究關注。隨著文字轉語音相關技術的發展，人們對於所產生語音的要求，除了語音內容的正確性以外，更要求所產生語音必須具有高自然度。而影響語音自然度的其中一個關鍵的因素為語音的語速。大多數早期所提出的文字轉語音模型是以自我迴歸模型為基礎，產生語音的方式為基於前一幀的語音內容，再接續產生下一幀的語音內容。然而，這種以自我迴歸模型產生語音的方式有一個最大的缺點，在於其對於所產生語音之語速缺乏控制能力。為了增加對於語速的控制能力，較後期所提出的文字轉語音模型便轉而利用非自我迴歸模型作為其模型基礎。以非自我迴歸模型作為基礎的文字轉語音模型，在訓練上所需要的資料量相當的大，因此在訓練上對於硬體設備的需求相當的高，所需的訓練時間也相對較長，造成模型訓練上的難度。因此，本論文提出一個調整文字轉語音模型所產生之語音語速之系統，本系統由一個文字對齊器、一個語速調整器、一個語音語速調整網路以及一個聲碼器組成。本系統的文字對齊器會找出語音中的文字邊界，語速調整器則會將語音轉換至頻域，接著根據文字邊界調整每個文字所對應語音之語速，調整的方式如下：加入空白幀以延長語音語速，刪除數個幀以縮短語音語速。調整後的頻譜則被輸入至語音語速調整網路，替空白幀內填入適當的語音內容，並弭平插入與刪除幀所造成幀與幀之間的不連續性。最後，再由聲碼器將調整後的頻譜轉換回時域，輸出成為語音音訊。實驗結果顯示透過本系統所調整語速之語音，與基於非自我迴歸模型的文字轉語音模型所座生之語音，其產生之語音品質相當，並相當接近真人錄製之語音品質。

關鍵字：**文字轉語音，音訊時長控制，蒙特婁文字對齊器**

In the past decades, synthesizing speech from texts, also known as Text to Speech (TTS), has drawn a great attention from researchers since it is applicable to a variety of applications. One of the factors that affects the prosody of synthesized speech is the speed at which it is spoken. Most of the TTS models proposed earlier are based on the autoregressive mechanism that generates speeches frame by frame. However, these autoregressive TTS models have a major drawback that lack the ability of controlling the duration of the synthesized speeches. In order to increase the ability of TTS models to control the duration of the synthesized speeches, many non-autoregressive TTS models are proposed to model the duration of synthesized speech. However, comparing with training an autoregressive TTS model, it takes a huge amount of data and computing power to train a non-autoregressive TTS model. Therefore, in thesis, a system for modifying the duration of speech synthesized from a TTS model is proposed. The proposed system consists of a forced aligner, a duration modifier, a neural network named ATM-Net and a vocoder. The forced aligner in the proposed system is adopted from the Montreal Forced Aligner with pretrained mandarin model. Once the boundary of each word / phoneme is determined, the duration modifier is then used to lengthen or shorten speech segments by inserting dummy frames or removing frames in a mel spectrogram respectively. The modified mel spectrogram is then fed into the ATM-Net to fill in audio contents as well as smoothen the discontinuities between frames. At last, the output mel spectrogram is used to synthesize the audio signal by a vocoder. Experiments show that the proposed system could modify the duration of speech and synthesize speech with natural prosody that is close to a real human does.

關鍵字：**Text-To-Speech，Audio Time-Scale Modification，Montreal Forced Aligner**

誌謝

感謝所有幫助過我的人。

內文目錄

表目錄

圖目錄

Chapter 1 Introduction

1.1 Motivation

In the past decades, synthesizing speech from texts, also known as Text to Speech (TTS), has drawn a great attention from researchers since it is applicable to a variety of applications. As more and more attention has been spent on the field of TTS, researchers start to aim at synthesizing speech that is not only intelligible but also expressive in the past decade. In order to make the synthesized speech expressive, a TTS model needs to generate speech with proper prosody in addition to correct semantic contents. One of the factors that affects the prosody of speech is the speed at which it is spoken. For example, a speaker would sound angry when a sentence is spoken faster. On the other hands, when a sentence is spoken at lower speed, the emotion of the speaker is commonly regarded as sorrow. Therefore, to control the duration of synthesized speech has become more popular in the field of TTS.

In general, TTS models can be classified into two major types, either an autoregressive [1, 2, 3, 4] or a non-autoregressive model [5, 6, 7, 8]. An autoregressive TTS model, as shown in Fig. 1.1, synthesizes speech in a sequence-to-sequence manner. Given a text as input, an autoregressive TTS model would generate frames in the synthesized speech sequentially, and the output frames would be fed-back to the autoregressive model to predict the next frame. The duration of each word in the input text is determined by an attention module and could not be changed easily. Therefore, an autoregressive model lacks the ability to control the duration of the output speech. On the other hand, in a non-autoregressive TTS model as shown in Fig. 1.2, an input text would be first encoded into some latent features. Then a duration predictor would be used to predict the duration of each frame in the latent feature. To control the duration of the synthesized speech, the major trend in the research field of TTS has been shifted from auto-regressive models to non-autoregressive models

with duration modeling. However, comparing with training an autoregressive TTS model, it would require a huge amount of data for the training of a non-autoregressive TTS model, and the cost of gathering such great amount of high-quality speech data are often extremely expensive. Besides, the training procedure of a non-autoregressive TTS model would also take a long period of time and relies on considerable computing power. Therefore, a system that can modify the duration of a speech at any specific region would help to relieve the enormous costs of training a non-autoregressive TTS model.

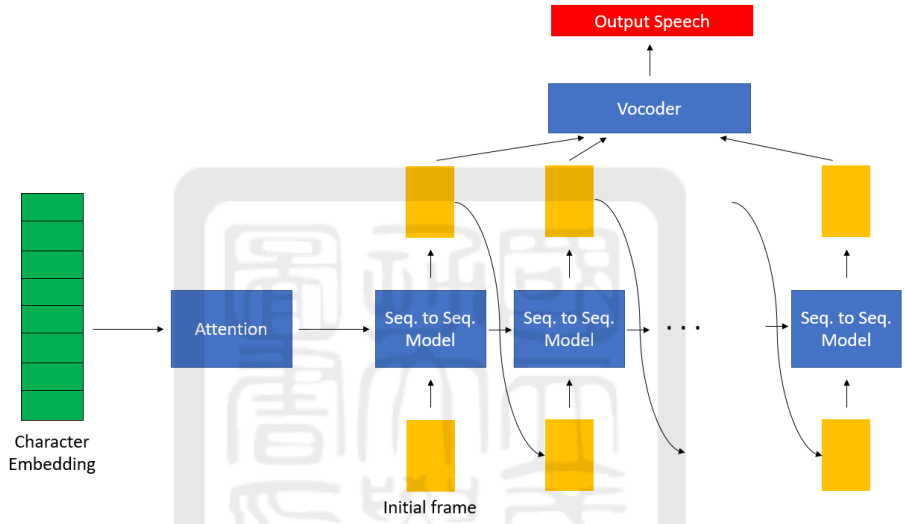


Fig. 1.1: The architecture of an autoregressive TTS

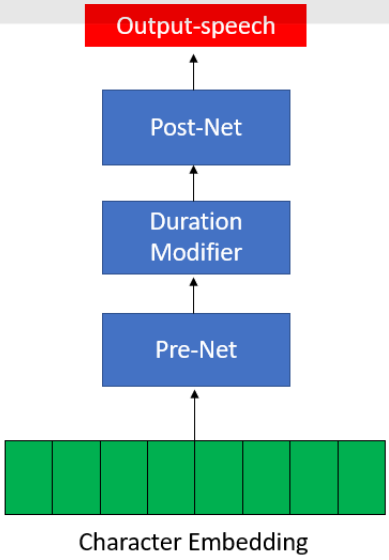


Fig. 1.2: The architecture of a non-autoregressive TTS

1.2 The Proposed Method

In this thesis, we proposed a duration control system that could modify the duration of a speech given the desired region to be modified, whether it is a word or even a phoneme, and the stretching ratio. The system takes a speech, and the corresponding text as input, and outputs a speech with its duration modified according to the desired stretching ratio, either speeding up or slowing down. The proposed system could modify different part of the synthesized speech with different stretching ratio. For instance, it could speed up the starting three words and slow down the last five words at the same time. This makes it effective in the context of modifying the duration of synthesized speech for expressing different prosody.

The system consists of a forced aligner, a duration modifier, an Audio Time-scale Modification Network, also known as ATM-Net, and a vocoder. Moreover, we proposed a two-stage training method for training the ATM-Net. The complete architecture of the proposed system can be seen in Fig. 1.3.

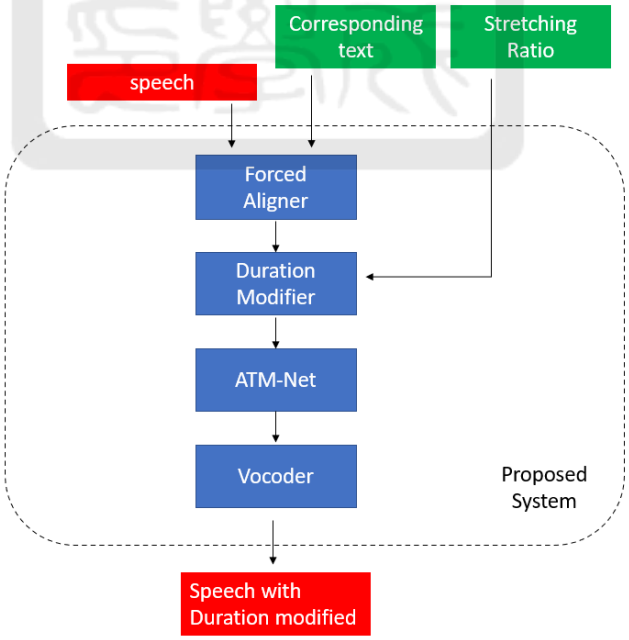


Fig. 1.3: The overall architecture of the proposed system

To precisely modify the duration of some region in the input speech signal, the boundary of the words in the input text must be determined first by the forced aligner. After that, the input speech signal is then transformed into a mel spectrogram, and a duration modifier is used to modify the duration of the mel spectrogram by inserting or deleting frames in the mel spectrogram. For a region to be lengthened, dummy frames are inserted among original frames. On the other hand, to shorten a specific region in a mel spectrogram, some of the frames at that region are removed. After that, the ATM-Net is used to reconstruct the mel spectrogram with its duration modified by filling in proper audio contents into the dummy frames and smoothen the discontinuities between frames that have been modified by the duration modifier. Finally, the mel spectrogram output from the ATM-Net is fed into a vocoder to synthesize a speech signal with its duration modified. More details of the proposed system will be articulated in Chapter 3.

1.3 An Overview of the Thesis

The remainder of this paper is structured as follows. In chapter 2 we will discusses related works in this research field. Chapter 3 describes the details of the proposed system. Experimental results of evaluating the proposed system as well as a brief discussion are described in Chapter 4. Chapter 5 is the conclusion of this thesis and the future ways of works.

Chapter 2 Related Works

2.1 Non-Autoregressive TTS

In the past decade, there are an abundance of research that manage to synthesize speech from text, also known as text to speech (TTS), using neural networks. Most of the earlier research on TTS utilizes attention mechanism to align an input text to the target output speech [1, 2, 3, 4]. These attention-based TTS models are also referred to as autoregressive TTS models.

However, as more and more focuses are placed on the prosody of the synthesized speech, the major drawback of these autoregressive TTS models have been revealed. That is, their inability to control the duration of the synthesized speech. The most prevalent method to control the duration in a TTS model is to use non-autoregressive TTS model in companion with a duration predictor instead of an autoregressive TTS model along with an attention mechanism. Generally, the text is usually much shorter than the corresponding speech signal. Consequently, a character, or a phoneme, will be expanded to multiple datapoints in the waveform, or multiple frames in the corresponding spectrogram or other acoustic features. The role of the duration predictor in a non-autoregressive TTS model is therefore used to determine the length ratio to be expanded.

There are several methods to build and to train a duration predictor in a non-autoregressive TTS model. In [5], Ren et al. proposed a TTS model with a duration predictor trained using the alignment result from an autoregressive TTS model [4]. The attention module in [4] was first used to generate the alignment relationship between texts and the corresponding speech signals. These data are then used to train the duration predictor used in [5]. Lancucki proposed a TTS model [9], which obtains the durations of the input text

from the attention matrix produced in Tacotron2 [10]. After the attention matrix is produced by the Tacotron2, the duration information can be obtained using the following formula:

$$d_i = \sum_{c=1}^{t}[argmax_f A_{f,c} = i] \quad (1)$$

Where d_i indicates the length in the output spectrogram that the ith input character should be mapped to. A is the attention matrix and t denotes the length of the output spectrogram. [11] uses the same duration predictor as that of in [5]. In addition, a method for jointly training the duration predictor with the TTS model is proposed. In [12], Baliaev et al. proposed a duration predictor trained with alignment data from an ASR model [13]. Despite the fact that the goal of an ASR model is in a complete opposite way compared with an TTS model, the similarity between a TTS model and an ASR model lies in that they are both trying to find a mapping between a text and a speech signal. Therefore, relationship similar to the attention matrix mentioned above can also be obtained from an ASR model and be used to train a duration predictor. The duration predictor in [14] is similar to that of in [9], moreover, Zeng et al. proposed a multi-phase training method for training the duration predictor using a mixed-density network. The duration predictor in [6] is trained using the Montreal Forced Aligner [15]. In [8], Yu et al. proposed a TTS model with a duration predictor consisting of 512-unit bidirectional LSTM layers and trained the duration predictor by minimizing the loss between predicted durations and the duration produced by a forced aligner. Non-Attentive Tacotron [16] proposed by Shen et al. consists a duration predictor trained using an unsupervised method. The data used to train the duration predictor is obtained from an aligner named FAVE proposed by Rosenfelder et al. [39]. Elias et al. proposed a TTS model in [7] with a duration predictor trained in parallel with the TTS model using lightweight convolution. [17] claims that previous duration predictor rounds the

duration before output to the next level, causing a rounding error. Besides, the rounding operation is non-differentiable, which cannot be propagated through the training process. Elias et al. introduced a duration predictor that is differentiable to address this problem. Abbas proposed a TTS network [18] that models the duration of the output speech at the phrase level.

These research, despite the fact that they could successfully model the duration of the synthesized speech, requires an enormous amount of data as well as great computing power to train their TTS models. Unlike these researches, the proposed system could be trained using a moderate amount of data and computing power to reach an appreciable result.

Table 2.1: The comparison of the non-autoregressive TTS models

Name of the TTS model	Type of the duration predictor	Other training methods
FastSpeech	From an autoregressive TTS model	None
FastPitch	From an autoregressive TTS model	None
Jdi-t	From an autoregressive TTS model	Jointly-training method
Talk-Net	From an ASR model	None
AlignTTS	From an autoregressive TTS model	A multi-phase training method
FastSpeech2	From a Speech Segmentation model	None

Durian	From a Speech Segmentation model	None
None-Attentive Tacotron	From a Speech Segmentation model	Unsupervised training method

2.2 Audio Time-Scale Modification

Time-Scale Modification, sometimes referred to as TSM, is a digital signal processing technique to modify the speed at which an audio signal is played and preserve the pitch of the audio signal at the same time. Because of a wide range to which TSM can be applied, there is a plenty of research digging into this field in the past few decades.

Allen et al. proposed an Overlapping-Add (OLA) method in [19] for time-scale modification by synthesizing waveform segments and concatenating them to the original audio signal according to the desired stretching ratio. In order to lengthen a signal, waveform segments are concatenated with less overlap with each other. On the contrary, to shorten a signal, waveform segments are concatenated with larger overlap with each other. However, the OLA method would lead to a serious problem that is called a phase distortion. The phase distortion is caused when one waveform segment is added to another without proper aligning the two waveform segments, resulting in a discontinuity in the output waveform. Second, simply adding one waveform segment to another without considering the amplitude the waveform segments would cause an increase in the total energy of the waveform. This would lead to an increase in the volume of the output waveform compared with the original one. To address the problem of phase distortion, Roucos et al. [20] proposed to synchronize frames by calculating cross-correlation between them before overlapping-add. Moreover, there are other researchers trying to synchronize frames using different techniques. For example, Verhelst et al. proposed to shift the target frame slightly and finding another frame sharing the highest similarity [36]. Rudresh et al. proposed an epoch-synchronization overlapping-add method in [21]. Unbiased correlation is calculated in [22] for synchronization. Other synchronization methods such as simplified normalized correlation in [23], envelope matching and modified envelope matching in [24] and [25] respectively,

and peak alignment in [26]. As deep neural networks have achieved a great success in audio processing, there are also attempts to address the TSM problem using a deep neural network. In [27], Xin et al. proposed a TSM method using a deep neural vocoder called HIFI-GAN [28]. They first interpolate the input mel spectrogram to a desired length, and then reconstruct the audio signal using HIFI-GAN.

The above-mentioned methods aim at modifying the duration of an audio as a whole. However, they are uncapable of modifying the duration of an audio to different ratio at different location in one speech, which is important in the context of generating speech with natural prosody. On the other hand, the proposed system could control the duration of speech to any ratio at any location in one speech.

Table 2.2: The comparison of the TSM algorithms

Name of TSM methods	Brief Descriptions
OLA	Simple windowing and overlapping
SOLA	Calculating the cross-correlation between frames
WSOLA	Slightly shifting frames
Autocorrelation OLA	Calculating the autocorrelation between frames
Speech-rate-controllable HifiGan	Frame interpolation

2.3 Speech Segmentation

In order to precisely modify the speed of speech at the desired region, one needs to determine the exact boundary of words or phonemes in a given speech. In [40], Brugnara et al. proposed to segment speech utilizing an Automatic Speech Recognition (ASR) trained using an Hidden Markov Model (HMM) and the Viterbi Algorithm [41]. The HMM ASR model is first used to build a set of acoustic-phonetic units corresponding to the input speech. Then the Viterbi algorithm is then applied to these acoustic-phonetic units to determine the word boundaries with the highest possibilities. Malfrere and Dutoit [42] proposed to segment speeches by comparing the original speech with the speech re-synthesized by a TTS model using the Dynamic Time Warping (DTW) algorithm. The text corresponding to the speech is first fed into a TTS model. Acoustic features of the synthesized speech as well as the original speech are then extracted respectively. Features of both speech and the original text are then aligned using the DTW algorithm to find the word segmentation in the original speech. Santen and Sproat proposed a segmentation method by detecting the frequency contour of the input speech and segment words by frequency changes [43]. Speech segments are categorized into several classes such as voiced-stop, voiced-affricates, voiced-fricatives, voiceless-stop, voiceless-affricates, voiceless-fricatives, nasals, liquids, glides, vowels, and so on. Phonetic boundaries are then determined by the frequency changes in the original speech. In [44], Katsamanis et al. proposed an alignment algorithm based on an ASR model. In their proposed algorithm, the ASR model will be re-trained in an iterative manner using the speech segments that can be welled aligned. They claimed that based on their iterative algorithm, the robustness of alignment results on long and noisy speeches would increase significantly. In [15], Mcauliffe et al. proposed the Montreal Forced Aligner (MFA) speech segmentation algorithm. The MFA is based on an HMM-GMM model with a three-staged training method. The MFCC of the speech data in the training corpus is first extracted with

the Cepstral Mean and Variance Normalization, CMVN [29]. In the first stage of training, a monophoneme HMM-GMM model is trained using the 39-dimensioned MFCC features. In the second stage of training, the training feature are grouped using the triphoneme-clustering to widen the vision of the HMM-GMM model. In the last stage of training, the MFCC feature are further transformed using the fMLLR [30] transformation for the training of the triphoneme model. This triphoneme model are then used for the alignment task. More detail of the training of the MFA model is depicted in Fig. 2.1. Since the MFA algorithm is the state-of-the-art in speech segmentation, in this thesis, we will adopt the MFA for the speech segmentation task in this thesis.

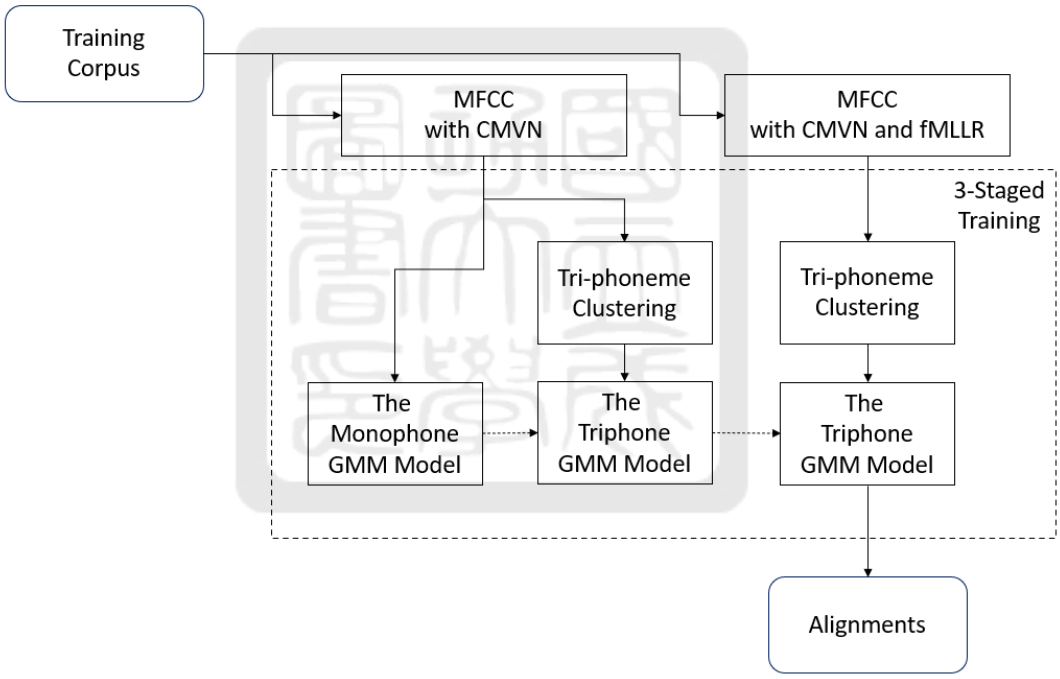


Fig. 2.1: The 3-staged training process of the MFA.

2.4 The Vocoder

Most of the acoustic engineering researches, such as Text-To-Speech (TTS) or Voice Conversion (VC), do not process speech signal directly. Instead, acoustic features are often calculated first. After some processing algorithms, the processed acoustic features should be fed into a vocoder in order to transform acoustic features back into speech signal that can be understood by human. In the past few decades, there are a number of research efforts being spent on vocoders.

Griffin and Lim proposed a vocoder named after both of their last names, the Griffin-Lim algorithm [38]. The concept behind the algorithm mainly lies in that an audio signal consists of both amplitude and phase information. Yet in a mel spectrogram, only the information of amplitude exists. The Griffin-Lim algorithm thus attempts to reconstruct the phase information needed for an audio signal by the following steps.

1. Convert a mel spectrogram to a magnitude spectrogram. This can be done by finding the Non-Negative Least Square (NNLS) solution for the mel spectrogram $\boldsymbol{S}$ given the mel bins $\boldsymbol{M}$ by the equation

$$argmin_X((MX)[i,j] - S[i,j])^2$$

Then the magnitude spectrogram will be the square root of $\boldsymbol{X}$.

2. Reconstruct the audio signal by iSTFT with the magnitude spectrogram from step 1 and a randomized phase information.

3. Do STFT to the audio signal reconstructed in step 2 to get both a magnitude spectrogram and the phase information. Discard the magnitude spectrogram and keep the phase information for the next iteration.

4. Reconstruct the audio signal by iSTFT with the magnitude spectrogram from step 1 and the phase information from step 3.

5. Iterate through step 2 to 4 until getting a satisfactory audio signal.

The procedure of the Griffin-Lim algorithm is depicted in fig. 2.2. Normally, the algorithm takes about 50 iterations to reach a satisfactory result.

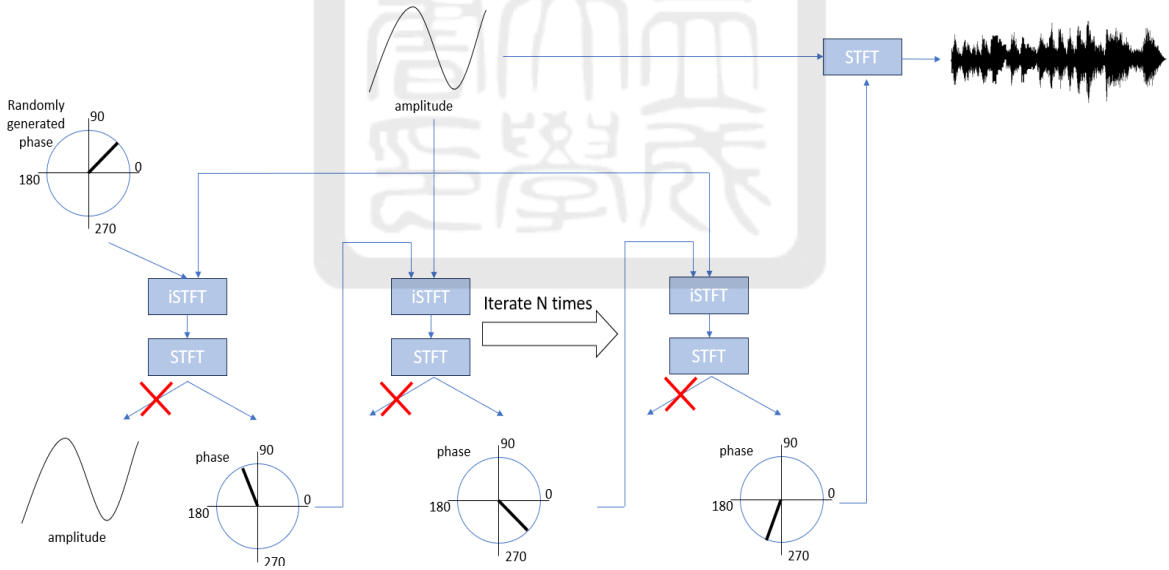


Fig. 2.2: The Griffin-Lim algorithm.

Another vocoder that is commonly used to synthesize speech signal is the WORLD vocoder proposed by Morise et al. [45]. The WORLD vocoder synthesizes speech signal using the fundamental frequency (F0), spectral envelope (SP) and aperiodic parameters (AP) of the original speech. The fundamental frequency of a speech is defined as the inverse of the smallest period of a periodic signal [45]. The WORLD vocoder captures the fundamental frequency of a speech using the DIO algorithm [46]. The spectral envelope of a speech is the contour of the peaks in its waveform. The CheapTrick algorithm [47] is used to calculate the spectral envelope of a speech. The aperiodic parameters of a speech represent the aperiodicity in the waveform and are estimated using the Platinum algorithm [48] in the WORLD vocoder. The WORLD vocoder can then synthesize speech signals using these features of the input speech.

With the advance in the research field of deep learning, several deep learning-based vocoders are also proposed. In [49], Prenger et al. proposed the Waveglow to synthesize speech signals through a flow-based network [50]. A flow-based network is well known for its capabilities of generating a complicated distribution from a simple distribution. In the case of the Waveglow, it generates speech signals by coupling the mel spectrogram of the input signal with a zero-mean Gaussian distribution.

Another type of deep generative network is the Generative Adversarial Network (GAN) [51]. A typical GAN is composed of a generator network and a discriminator network. The goal of a generator network is to generate fake data as real as possible. On the other hand, the discriminator network is responsible for discriminating real data from the ones generated by the generator network. Under this adversarial training schema, the data generated by a generator network is expected to be as real as possible. A typical training schema of a GAN

is shown in fig. 2.3.

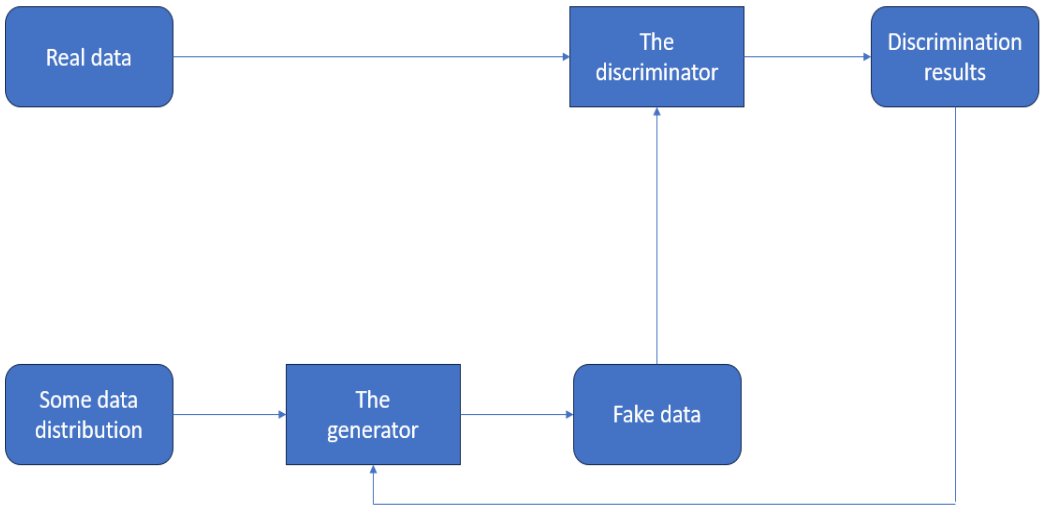


Fig. 2.3: The training schema of a GAN. The generator generates fake data from some data distribution. The discriminator discriminates the fake data from real data. The discrimination results are then feedback to the generator to refine the generator.

There are also vocoders that are based on the concept of the GAN. In [52], Yamamoto et al. proposed the WaveGAN that generates speech signals from the corresponding mel spectrogram and a random Gaussian noise. In [53], Kumar et al. proposed a generator that is composed of layers of deconvolusion layers to up-sample low resolution mel spectrogram back into speech signal of high resolution. Mel spectrogram of low resolution are chosen as the input of the MelGAN to reduce the complexity of computation. The architecture of the generator of the MelGAN is shown in fig. 2.4. Afterwards, Yang et al. further improved the architecture of the MelGAN in [33], named as the VocGAN. The main difference between the MelGAN and the VocGan lies in that the VocGAN uses a hierarchically-nested discriminator that discriminate audio signals at different stages of scaling up a mel spectrogram to an audio signal. More detail of the VOC-Gan are shown in Fig. 2.5. The

VOC-Gan has been shown to perform quite well in reconstructing audio signals. Therefore, we adopt the VOC-Gan as the vocoder in our proposed system.

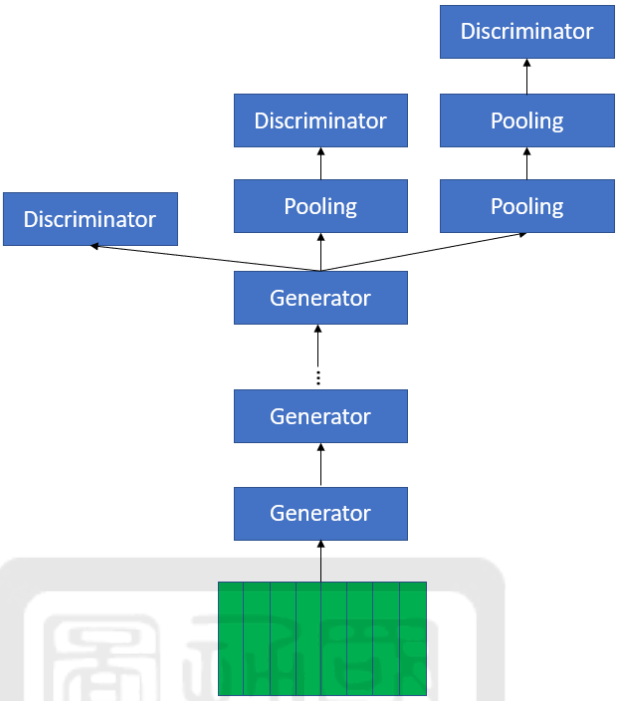


Fig. 2.4: The architecture of the MelGAN.

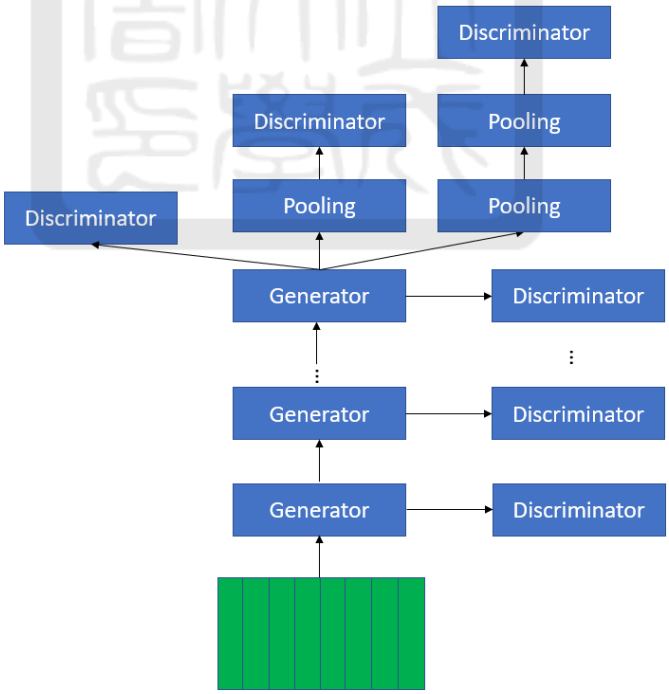


Fig. 2.5: The architecture of the VocGAN.

Chapter 3 The Proposed System

In this thesis, a system that can modify the duration of an audio signal synthesized by a TTS model is proposed. The system can be further divided into the following components: a forced aligner, a duration modifier, an **A**udio **T**ime-Scale **M**odification **Net**work (ATM-Net) and a vocoder. The system takes the synthesized speech from an TTS model, the corresponding text, and the desired stretching ratio as input, and outputs the speech with its duration modified according to the stretching ratio. The minimum unit for stretching is a word or even a phoneme. For example, for a sentence {我, 1.2}{不想吃, 0.8}洋芋片, indicating that the duration of the first word should be lengthen 1.2 times while the duration of the phrase 不想吃 should be shorten to 0.8 times of the original length.

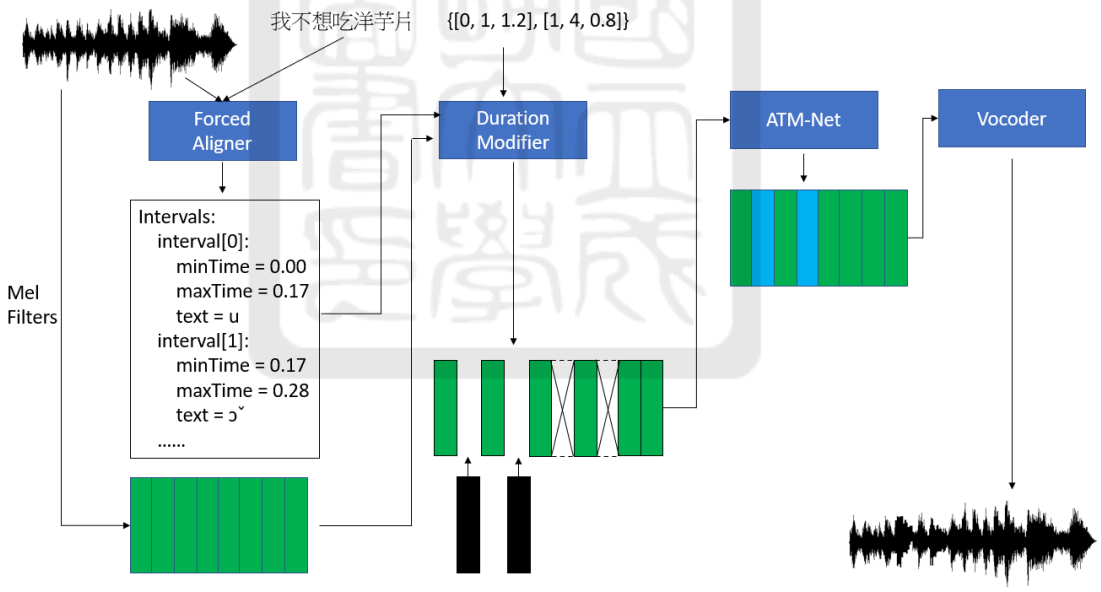


Fig. 3.1: A complete overview of the workflow of the proposed system. The input audio signal is transformed into mel spectrogram (as shown in green). The duration modifier inserts dummy frames (as shown in black) or deletes frames (shown in dotted lines). The ATM-Net fills the dummy frames with proper audio contents (as shown in blue).

The proposed system modifies the duration of the speech in the following steps. First, the forced aligner would find the position and the length of each phoneme in the speech and produce a speech with timestamps indicating the position and the length of each phoneme. Second, the speech will be transformed into a mel spectrogram, and the mel spectrogram will be lengthen/shorten according to the stretching ratio by the duration modifier. For the case of lengthening, dummy frames will be inserted amid the original frames. On the other hand, frames will be deleted for the case of shortening. Third, the mel spectrogram with dummy and deleted frames will be fed into the proposed ATM-Net. The goal of the ATM-Net is to fill in the dummy frames with proper audio contents and smoothen the discontinuity between modified and original frames. Finally, the output mel spectrogram of the ATM-Net will be fed into a vocoder to generate speech with duration modified. A complete workflow of the proposed system is shown in Fig. 3.1. More detail of the components in the proposed system is described below.

3.1 The Forced Aligner

In order to modify the duration of each word in the speech with different ratio, we need to determine the boundary of each phoneme in a speech. In other words, we need to find the exact timing that each phoneme starts and ends in a speech. The task to find the starting and ending time of each phoneme in a speech can be done by a forced aligner. In this thesis, we adopt the Montreal Forced Aligner (MFA) [15] to determine the boundaries of words in a speech.

We use the pretrained model for mandarin released by the official developers for the alignment task [31]. The MFA takes an audio together with the corresponding transcript as input, breaks the speech down into phonemes, and finally outputs the starting and ending time of each phoneme in the speech. An example of the output of and MFA is shown in Fig. 3.2. Take the first interval as an example, the interval starts at 0.00 second in the waveform and ends at 0.17 second. The phoneme spoken in this audio interval is ‘u’. And so on. With the help of the MFA, we could break down an input speech into words or even phonemes, with timestamps specifying the starting and ending times of them. This information is then fed into the duration modifier for duration modification.

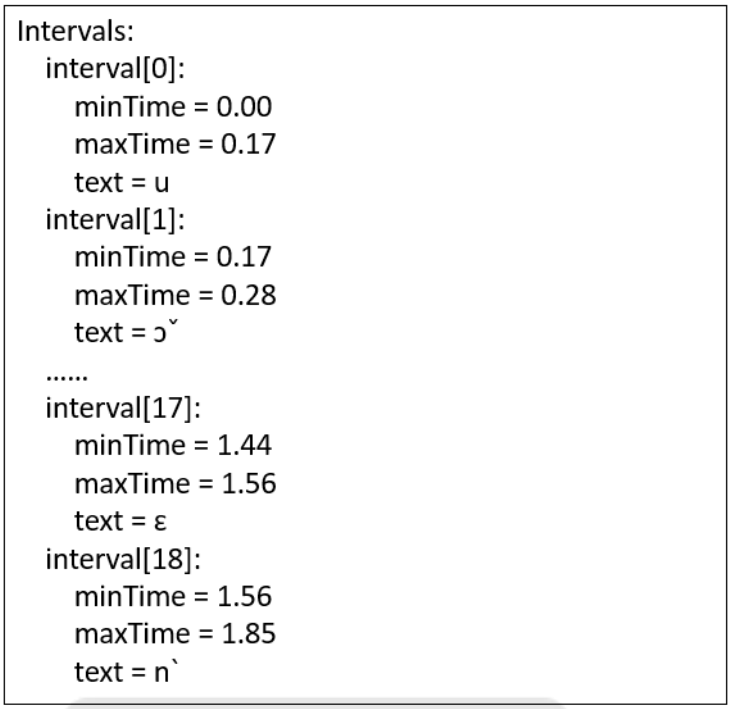


Fig. 3.2: An example output of the MFA . It can be seen that the speech is segmented into 18 intervals. The starting time, denoted as minTime, and the ending time, denoted as maxTime, and the corresponding text of this speech interval are listed in each of the interval.

3.2 The Duration Modifier

To change the duration of speech, the duration modifier is used in the proposed system. The duration modifier changes the duration of each word in the speech according to the input stretching ratio. Before adjusting the duration of the speech, we first transform the speech into its corresponding mel spectrogram.

A spectrogram is an acoustic feature that captures the time, frequency and amplitude of a speech signal. For a speech signal with sampling rate S, the spectrogram can be obtained by filtering the speech signal with a set of filters with central frequencies ranging from 0 Hz to S / 2 Hz. However, the perception of human hearing on frequencies are not distributed linearly across the range perceptible by the ears. If the central frequencies of the set of filters are linearly distributed, they might not be able to represent the speech signal properly. Therefore, Stevens et al. proposed a non-linear frequency scale that are closer to the frequency distribution of human hearing in [54], so called the mel scale. The relationship between a linear frequency scale and a mel frequency scale are shown as follows.

$$f_{mel} = 2595 \log_{10}(1 + \frac{f_{linear}}{700})$$

A spectrogram created using the set of filters with their central frequencies distributed in a mel scale is then called a mel spectrogram. As shown in fig. 3.3.

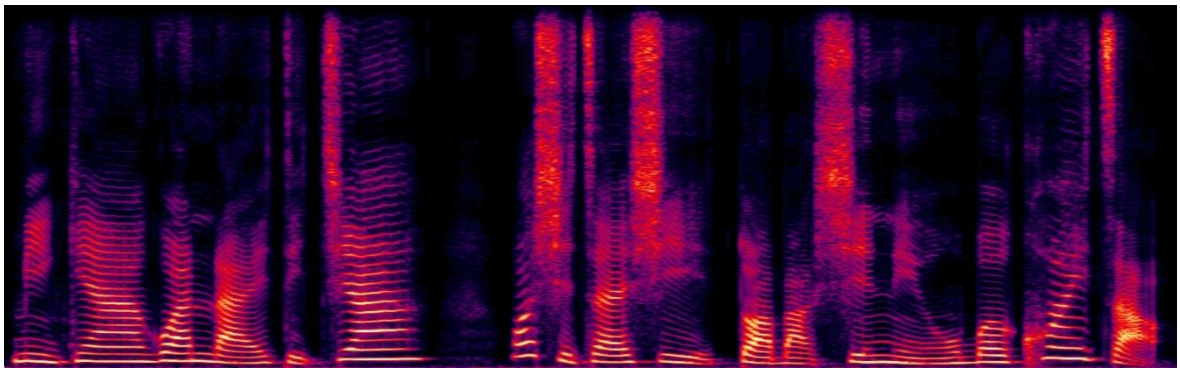


Fig. 3.3: A mel spectrogram. The x-axis is the time, and the y-axis represents the frequency. And the brightness represents the magnitude of the speech signal.

After the speech signal is converted into a mel spectrogram, it is then ready to be lengthened or shortened. In the case of lengthening a word, the duration modifier inserts dummy frames between original frames in the region corresponding to the specific word given by the MFA to fit the stretching ratio. A dummy frame is a frame with its value all equals to zero. For instance, to double the duration of a certain word, the duration modifier inserts dummy frames the same number as the frames corresponding to the word. In this thesis, the insertion strategy of dummy frames is to insert dummy frames between original frames of the word, as shown in Fig. 3.4. On the other hand, in the case of shortening a word, the duration modifier removes frames belonging to the word to fit the stretching ratio. For instance, to half the duration of a certain word, the duration modifier removes half of the frames corresponding to the word. The removing strategy is also to remove frames in the given region uniformly, as shown in Fig. 3.5.

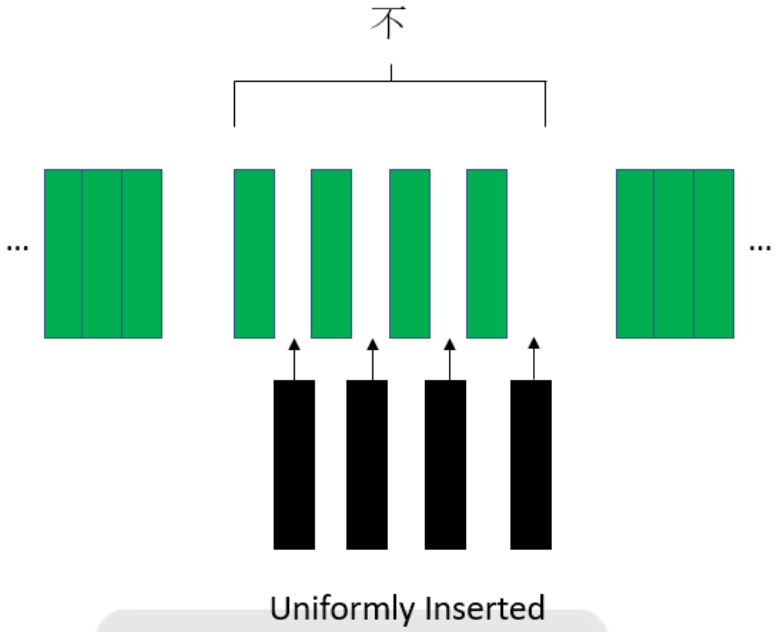


Fig. 3.4: Inserting dummy frames into the mel spectrogram. Dummy frames, shown in black, are inserted at the desired region between the original frames at that region.

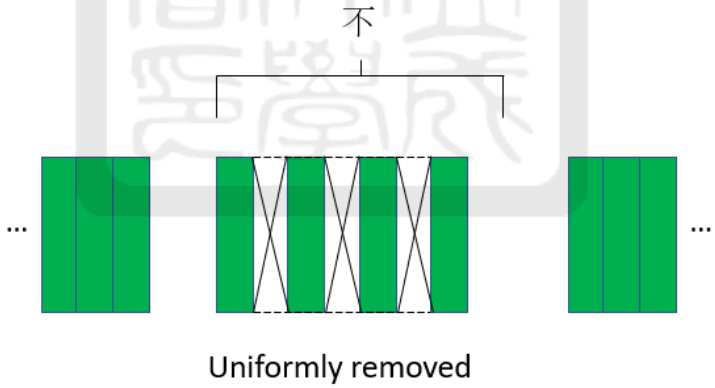


Fig. 3.5: Removing frames from the mel spectrogram. Removed frames, shown in dotted lines, are selected uniformly at the desired region.

3.3 The ATM-Net

3.3.1 The Model Architecture

The output of the duration modifier is a mel spectrogram that fits the desiring stretching ratio. However, audio contents must be filled into the dummy frames. Meanwhile, discontinuity appears between dummy and original frames, and between frames that was originally separated but the frames amid them are deleted. To fill in the inserted dummy frames with proper audio contents and to smoothen the discontinuity between frames, we proposed an ATM-Net in this thesis. The architecture of the ATM-Net is adopted from the Spectrogram Super-resolution Network (SSRN, Fig. 3.8) proposed in [32] with a slight modification.

Initially, the SSRN is proposed as a supplementary network to help reducing the complexity of training an TTS network. It is designed to up-sample an input mel spectrogram to an output magnitude spectrogram by an up-sampling factor, which equals to 4 in the paper that the SSRN is originally proposed [32]. With the help of the SSRN, the TTS model can then reduce the dimension of its output. For instance, if the SSRN can up-sample the mel spectrogram by 4 times, it means that the output dimension of the TTS network can be 4-times reduced, and thus lower the computational efforts of the TTS model under the same hardware circumstance.

In order to adopt the SSRN in this paper to fulfill the goals mentioned earlier in this chapter, three modifications need to be made to the SSRN. First, the original SSRN uses stacks of deconvolution layers to up-sample the input mel spectrogram to the desired up-sampling factor. Nonetheless, in this thesis, the modification of the length of the input mel spectrogram is already done by the duration modifier introduced in the last chapter by

inserting dummy frames and removing unwanted frames at desired regions in the input mel spectrogram. Once the mel spectrogram is output by the duration modifier, the length of the mel spectrogram is determined and should not be changed by the ATM-Net. The deconvolutional layers that is originally in the architecture of the SSRN are thus removed in the proposed ATM-Net.

Second, the 1D convolutional layers [59] in the original SSRN are followed by the highway convolutional blocks [60]. The architecture of a highway convolutional block is shown in fig. 3.6. The highway convolutional blocks are proposed by the Srivastava et al. and can be denoted by formula 2.

$$y = H(x, W_H) \cdot T(x, W_T) + x \cdot C(x, W_C) \quad (2)$$

Where H, T and C are all non-linear functions and W_H, W_T and W_C are the weight of them respectively. For simplicity, C is often defined as 1 – T, then formula 3 can be re-written as

$$y = H(x, W_H) \cdot T(x, W_T) + x \cdot (1 - T(x, W_T)) \quad (3)$$

Finally, if T(x) is defined as a sigmoid function, then formula 4 can be further re-written as

$$y = \begin{cases} x, & if\ T(x, W_T) = 0 \\ H(x, W_H), & if\ T(x, W_T) = 1 \end{cases} \quad (4)$$

That is, when $T(x, W_T) = 0$, the input x by-passed to the output. On the other hand, when $T(x, W_T) = 1$, the input x is passed to the output through a non-linear function $H(x, W_H)$.

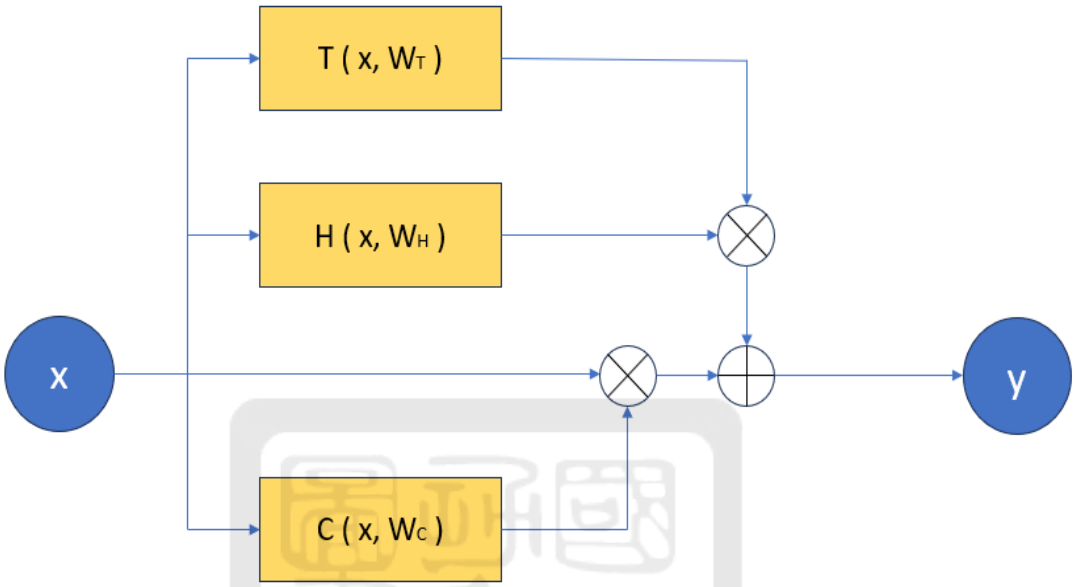


Fig 3.6: A highway convolutional blocks. For simplicity, C is often defined as 1 – T and T is often defined as a sigmoid function.

However, the highway convolutional blocks are reported to suffer from the problem of degradation [55]. That is, as the depth of the networks grows deeper, the performance of the network gets saturated. This is because the output of a convolutional layer are often a small number, much smaller than 1 for most of the time. Consequently, as the layers are stacked deeper and deeper, the value output by each layer will become smaller and smaller, and finally converges to zero. This phenomenon is so called as a vanishing gradient [55]. Therefore, in this thesis, the highway convolutional blocks are replaced by the residual blocks proposed in [55]. A residual block can be denoted as follows.

$$y = H(x, W_H) + x \quad (5)$$

Multiple residual blocks can be cascaded to form a deeper network without suffering from the problem of degradation, making it a simple yet effective component that is commonly used in many of the networks [find 3 using resnet]. The architecture of a residual block is shown in fig. 3.7.

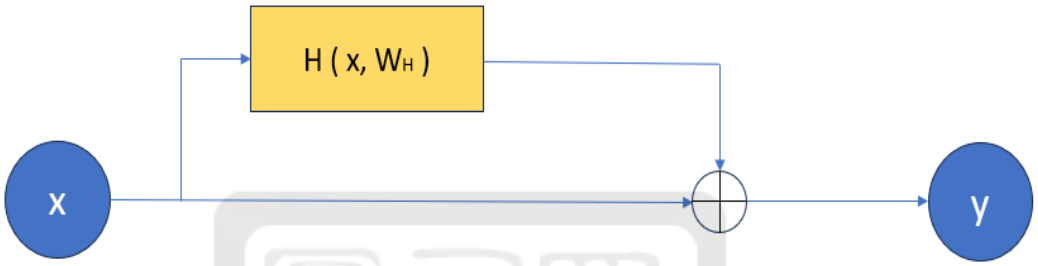


Fig. 3.7: A residual block.

Third, since the output of the SSRN is fed to the Griffin-Lim algorithm to synthesize speech signal, the dimension of the output layer of the SSRN is set to be 1025, which is equal to the dimension of a magnitude spectrogram. However, the output of the ATM-Net is used to synthesize a speech signal by the VocGAN. And the input of the VocGAN is a mel spectrogram with a dimension 80. Therefore, the dimension of the output layer of the ATM-Net is set to be 80. The overall architecture of the proposed ATM-Net is shown in Fig. 3.9.

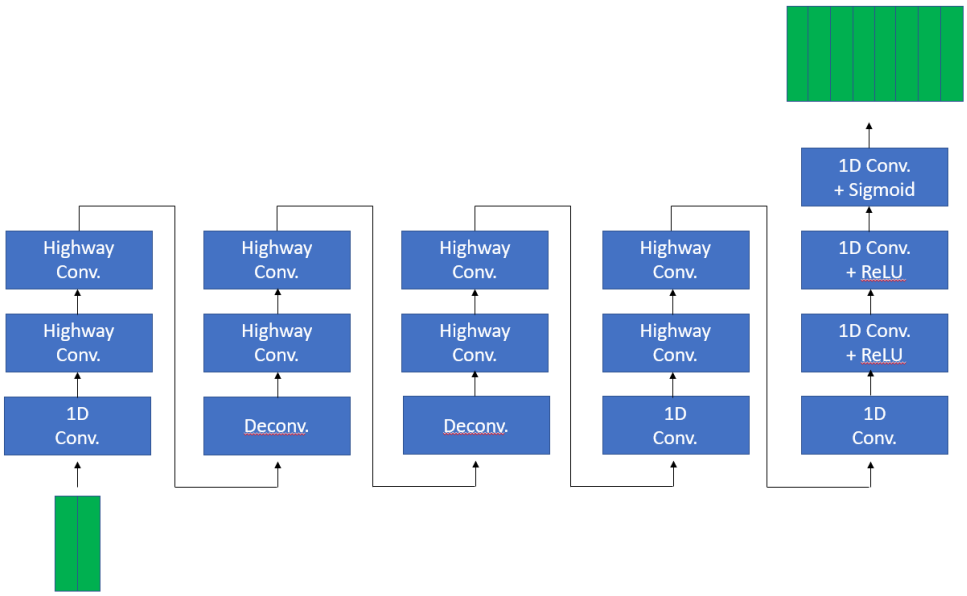


Fig. 3.8: The model architecture of the SSRN. The SSRN would restore a mel spectrogram reduced in time scale back to a full mel spectrogram.

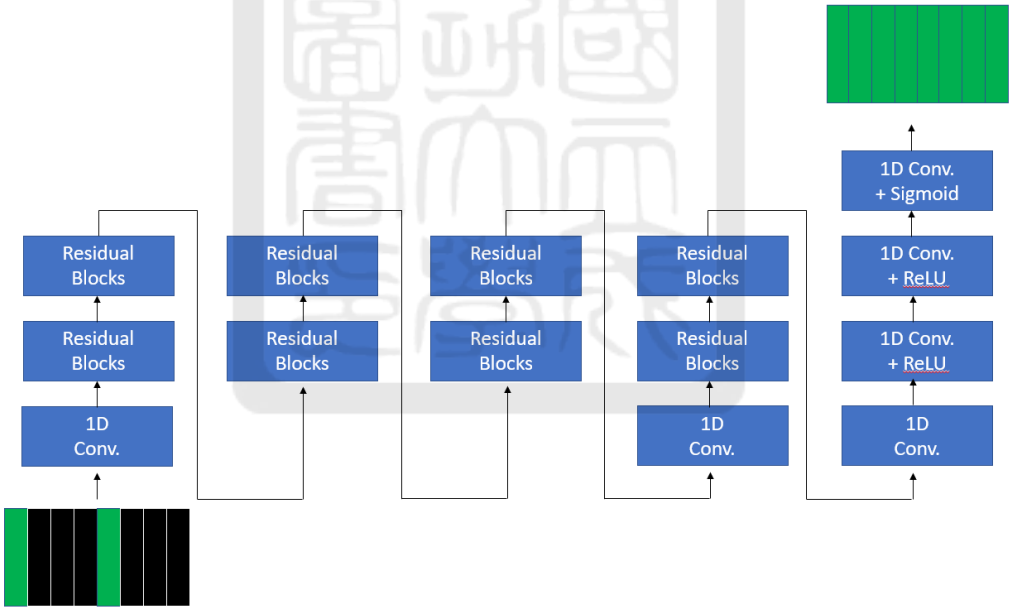


Fig. 3.9: The model architecture of the ATM-Net. The ATM-Net aims to fill the masked frames in the mel spectrogram with proper audio contents, and outputs a full mel spectrogram.

3.3.2 The Two-Staged Training Method

In the above paragraph, the architecture of the proposed ATM-Net is introduced. The goal of the proposed ATM-Net is as follows: 1) To fill proper audio contents into the dummy frames inserted by the duration modifier. 2) To smoothen the discontinuity between modified and original frames. To achieve these goals, we proposed a two-stage training method of the ATM-Net.

At stage 1, we train the ATM-Net as an encoder/decoder pair. Given an input mel spectrogram, the ATM-Net is trained to reproduce a mel spectrogram that is exactly the same as the input one. The learning objective is to minimize the difference between the input and the output mel spectrogram using the following loss function.

$$Loss = \sum \left| Mel_{gt} - Mel_{pred} \right| \quad (6)$$

After this stage, the ATM-Net can be seen as an encoder/decoder pair for the mel spectrogram. Given an input mel spectrogram, the ATM-Net should be able to reproduce the same mel spectrogram as the input one. In other words, given a mel spectrogram with dummy frames inserted amid the original frames, the ATM-Net will treat the dummy frames the same as the original frames and output a mel spectrogram with dummy frames at the same location. In order to train the ATM-Net to fill in correct audio contents in the dummy frames, the ATM-Net thus need to be further fine-tuned.

Therefore, in the second stage, a mel spectrogram with some of its frames masked are given as input to the ATM-Net. In order to mask some frames in a mel spectrogram, we multiply the mel spectrogram by a masking matrix *M*. The masking matrix *M* is a matrix

with exact the same dimension as that of the mel spectrogram except for some of its columns to be all zero. The intuition behind training the ATM-Net with masked frames is that if the ATM-Net sees a dummy frame inserted by the duration modifier among two original frames in a mel spectrogram, it has to fill in the dummy frame with proper audio contents, making the spectrogram a normal spectrogram with its duration lengthened. The masked frames in the training data in this stage are thus used to mimic the dummy frames inserted, and the ATM-Net should learn to fill in the masked frames with proper audio contents to make the output mel spectrogram sounds as real as possible. This is done by minimizing the loss between the output mel spectrogram and the unmasked input mel spectrogram.

Moreover, two masking strategies are proposed for training our ATM-Net. The first strategy is to mask the input mel spectrogram with a uniformly distributed mask. In other words, the masked frames are all separated with the same distance between each other. Another strategy is to use a randomly distributed mask on the input mel spectrogram. That is, the masked frames are separated with each other in random distances. The two masking strategies are depicted in fig. 3.10.

(a) (b)

Fig. 3.10: Two masking strategies. (a), uniformly masking strategy and (b), randomly masking strategy.

Finally, the ratio between the number of masked and unmasked frames is defined as the masking ratio. The masking ratio is also an experimental variable, and its impact on the performance of our ATM-Net will also be evaluated in the next paragraph. The proposed two-stage training method are depicted more detailly in Fig. 3.11.

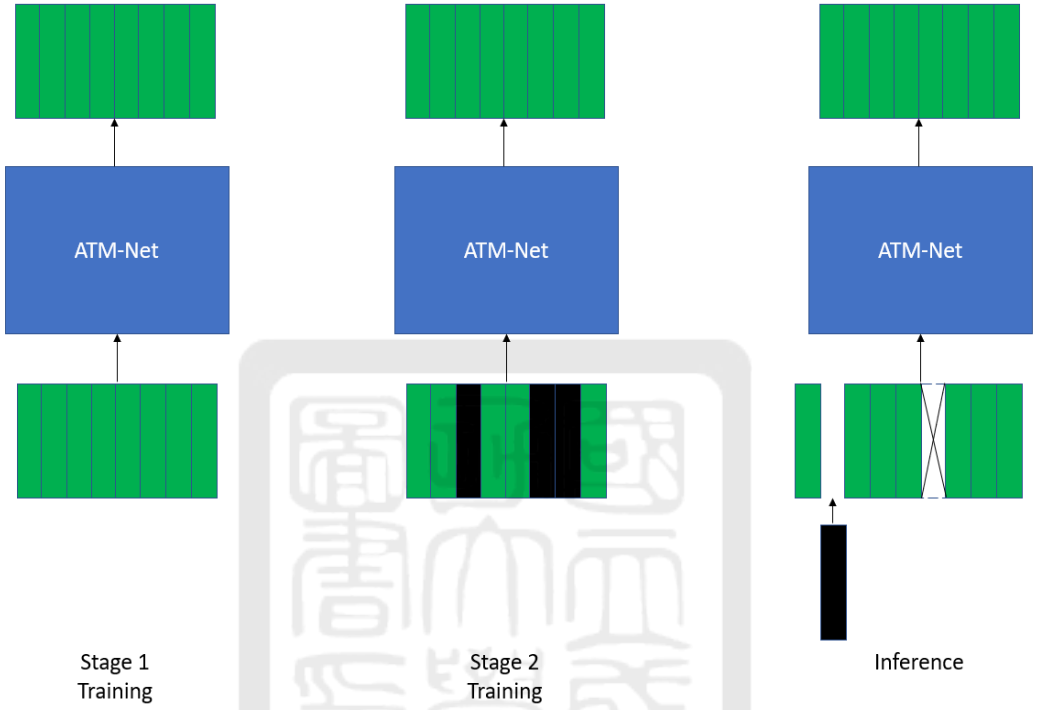


Fig. 3.11: The two-staged training method of the ATM-Net.

3.4 The Vocoder

After the mel spectrogram with duration modified are produced, the final step of the proposed system is to convert the mel spectrogram back into an audio signal. The task to convert a mel spectrogram to an audio signal is usually done by a vocoder. In this thesis, The VocGAN [33] is used for vocoding a mel spectrogram back into a speech audio signal.

In order to train a VocGAN for mandarin, a pre-trained weight of the VocGAN is obtained from [56]. The pre-trained weight is trained using the LJSpeech dataset [57], a dataset containing 13100 speeches from a woman speaker. And the weight is then fine-tuned using two datasets of a man and a woman mandarin speaker respectively. Details of both mandarin datasets will be articulated in the next chapter.

Chapter 4 Experimental Results

4.1 Data Preparation

Speeches synthesized by a TTS model are used to evaluate the effectiveness of the proposed system. The DC-TTS [32] proposed by Tachibana et al. are adopted in this thesis to synthesize speeches for duration modification. The reason why the DC-TTS model are chosen is that it is an autoregressive TTS model and could be trained with a moderate amount of data in comparison with other non-autoregressive TTS models. We train the DC-TTS model using both a male and a female dataset, named as M01 and F01 respectively, to evaluate the proposed system. Both datasets contain 1560 audio files recorded in a professional studio, and the corresponding transcripts. The data structure of the datasets and the snapshot of the transcript are shown in fig. 4.1 and 4.2 respectively. The audio files are in the format of .wav, sampled at 22050 Hz. Before being input to the proposed system, the voice data is first transformed into mel spectrograms with an 80-bin mel-scale filter set. Both datasets are split into a portion of 0.85 / 0.1 / 0.05 for training / validating / testing. More statistics of both datasets are listed in table 4.1. The training process takes approximately 6 ours for a single dataset on a PC with 2 2080 Ti graphic cards, with 12 GB memories each. The batch size is set to be 32 and the learning rate is 0.00001. An early-stop mechanism is applied if the loss does not drop for five consecutive training epochs after 10K epochs of training.

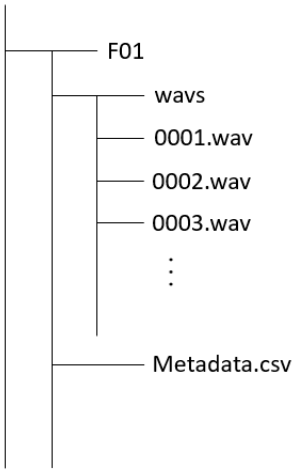


Fig. 4.1: The data structure of the datasets. The F01 dataset is taken as an example.

	filename	text	bopomo
2	0001.wav	從來沒有一個欺善怕惡的人有好下場	ㄘㄨㄥˊ ㄌㄞˊ ㄇㄟˊ ㄧㄡˇ
3	0002.wav	山下智久沒有出席	ㄕㄢ_ㄒㄧㄚˋ ㄓˋ ㄐㄧㄡˇ ㄇㄟˊ
4	0003.wav	把雞胸肉靜置二十分鐘	ㄅㄚˇ ㄐㄧ_ㄒㄩㄥ_ㄖㄡˋ ㄐㄧㄥˋ

Fig. 4.2: The transcript of the audio files.

Table 4.1: The statistics of the datasets

	F01	M01
Number of Sentences	1560	
File Format	.wav	
Sampling Rate	22050	
Total Length	4761 seconds	4803 seconds
Average Length	3.05 seconds	3.08 seconds
Average Word per Sentence	13.61 word	

4.2 Experimental Results

We use the **M**ean **O**pinion **S**core (MOS) [34] to evaluate the audio signal with its duration modified by our system. The MOS is the most common scale used to measure the naturalness of generated audio signals by an TTS model. The higher a score is, the more a speech sounds like spoken by a real human. The scoring criteria is shown in Table 4.2 [34]. In this experiment, the MOS will be the averaging score of 10 randomly selected listeners after listening to the testing audios. Moreover, in order to validate the effectiveness of the MOS, the Cohen's Kappa Coefficient [61] of the MOS given by the listeners are also calculated. The Cohen's Kappa Coefficient is a commonly used statistics to verify the agreement between different testers, and a higher value of the Cohen's Kappa Coefficient indicates the higher degree of agreement reached by all the testers. Conventionally, a Cohen's Kappa Coefficient higher than 0.75 shows that all the testers have reached a high agreement. We define the two scores are in the same class if the difference between them is less or equal to 1. Otherwise, the two scores are regarded as not in the same class.

The testing audios includes audios from real human and audios synthesized from a TTS model and then with their duration modified by our system. The sequence of playing the audios is randomly shuffled. Two other algorithms that could modify the speed of a speech signal are adopted in comparison with the proposed ATM-Net in this thesis. The first algorithm chosen for comparison is the WSOLA algorithm [36]. Despite that it was proposed in 1993, it is still the most commonly used TSM algorithm in several audio editing software [37, 58]. The next algorithm adopted is the algorithm proposed in [27]. By utilizing the power of the deep neural networks, it has reached the state-of-the-art performance in modifying the speed of a speech signal. We will evaluate the proposed system from four aspects. 1, the naturalness of the speech shortened by the proposed system. 2, the naturalness

of the speech lengthened by the proposed system. 3, the naturalness of the speech with its duration partially modified. And 4, effects that using different vocoders in the proposed system.

Table 4.2: The scoring criteria of the MOS evaluation method.

Score	Quality	Listening Effort Scale
1	Excellent	No effort required
2	Good	No appreciable effort required
3	Fair	Moderate effort required
4	Poor	Considerable effort required
5	Bad	No meaning understood with reasonable effort

4.2.1 Shortening the Duration of an Audio Signal

To shortening the duration of an audio, we use the duration modifier to delete frames at the region to be shorten. The goal of this experiment is to evaluate the overall quality of the audio signals shortened by the proposed system, either trained by randomly-mask or uniformly-mask strategies. Audio signals will be shortened as a whole by three different degrees, ranging from 3 / 4 of the original time, 2 / 3 of the original time and half of the original time respectively. The masking ratio is set to 0.1, 0.5 and 0.9. The result can be seen from Fig. 4.3 to 4.5.

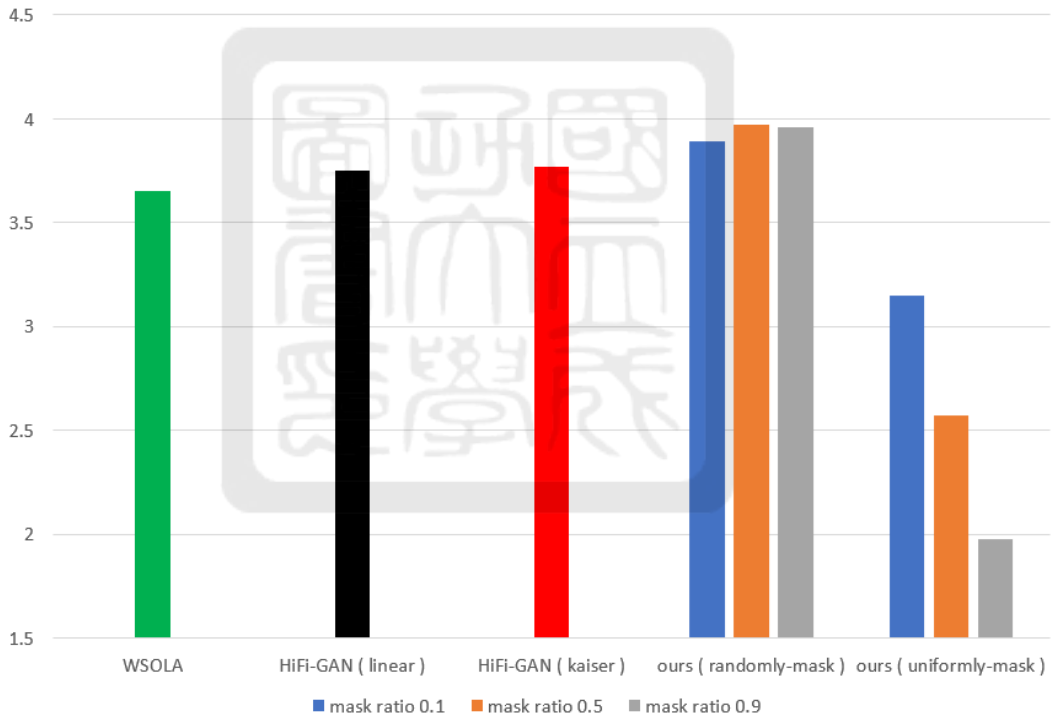


Fig. 4.3: The MOS of speech shortened by WSOLA and the proposed method. The speech is shortened to 3 / 4 of the original time.

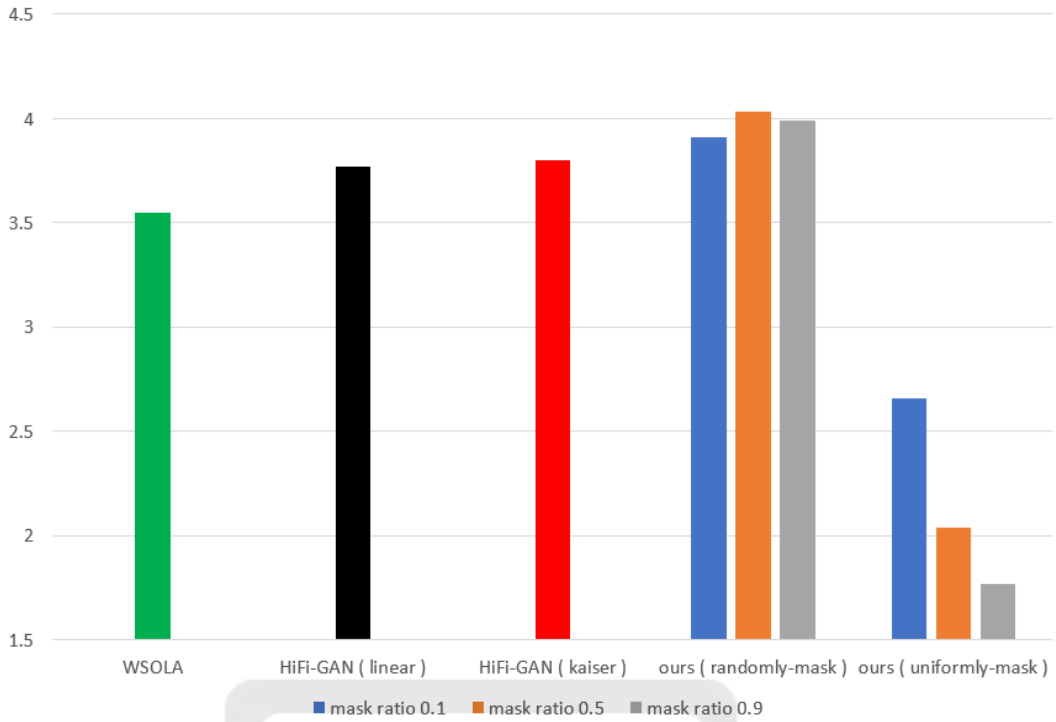


Fig. 4.4: The MOS of speech shortened by WSOLA and the proposed method. The speech is shortened to 2 / 3 of the original time.

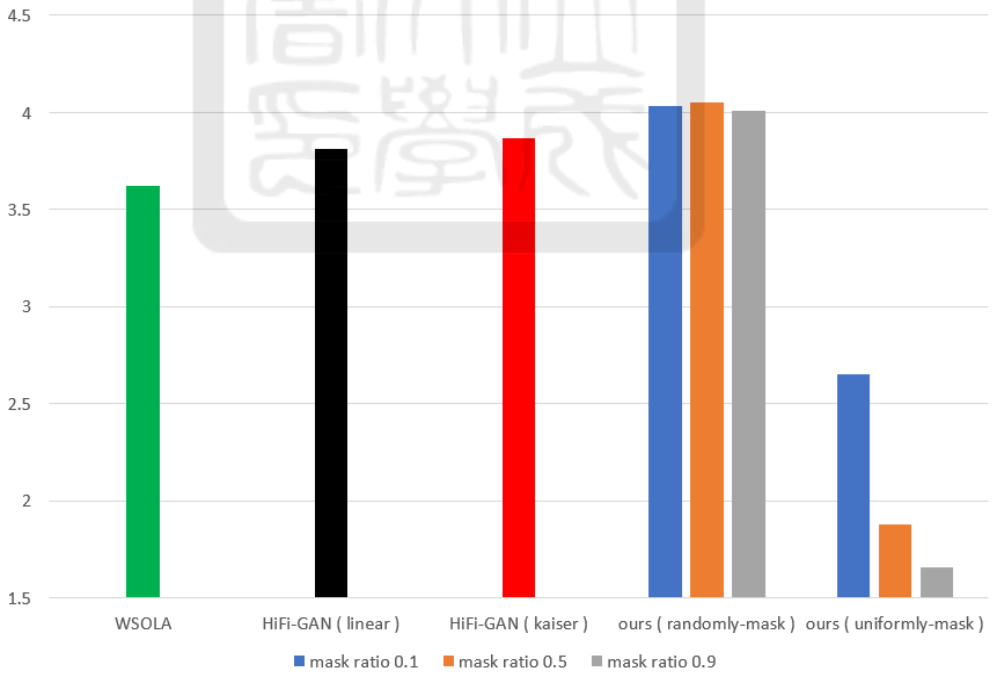


Fig. 4.5: The MOS of speech shortened by WSOLA and the proposed method. The speech is shortened to half of the original time.

4.2.2 Lengthening the Duration of an Audio Signal

To lengthening the duration of an audio, we use the duration modifier to insert dummy frames between original frames at the region to be lengthen. Then, the ATM-Net would fill the dummy frames with proper audio contents. Audio signals will be lengthened as a whole by three different degrees, ranging from 5 / 4 of the original time, 4 / 3 of the original time and 3 / 2 of the original time. The masking ratio is set to 0.1, 0.5 and 0.9. The result can be seen in Fig. 4.6 to 4.8.

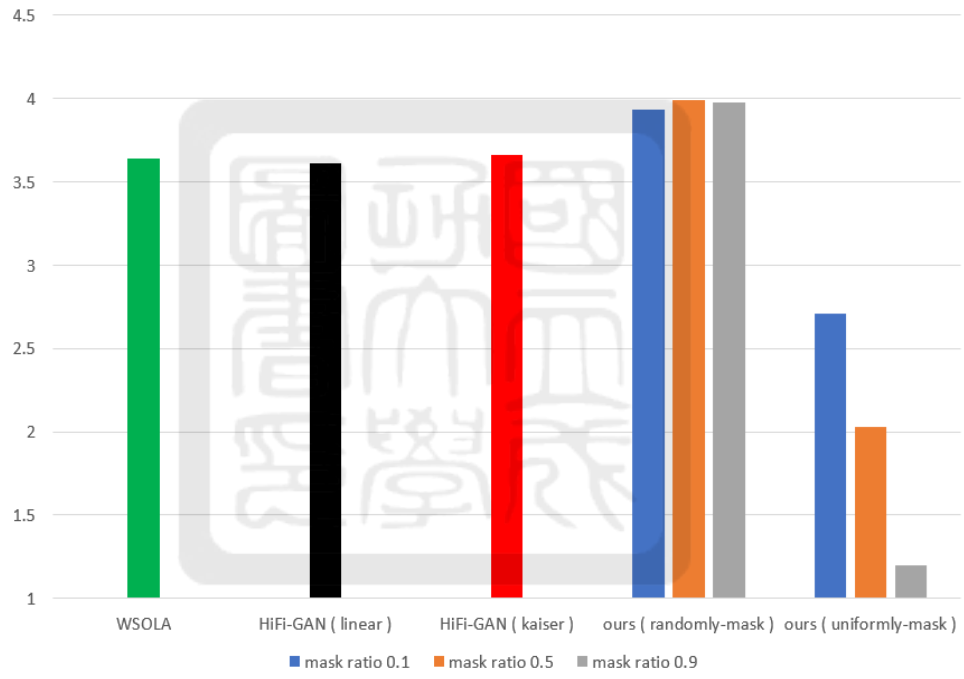


Fig. 4.6: The MOS of speech lengthened by WSOLA and the proposed method. The speech is lengthened to 5 / 4 of the original time.

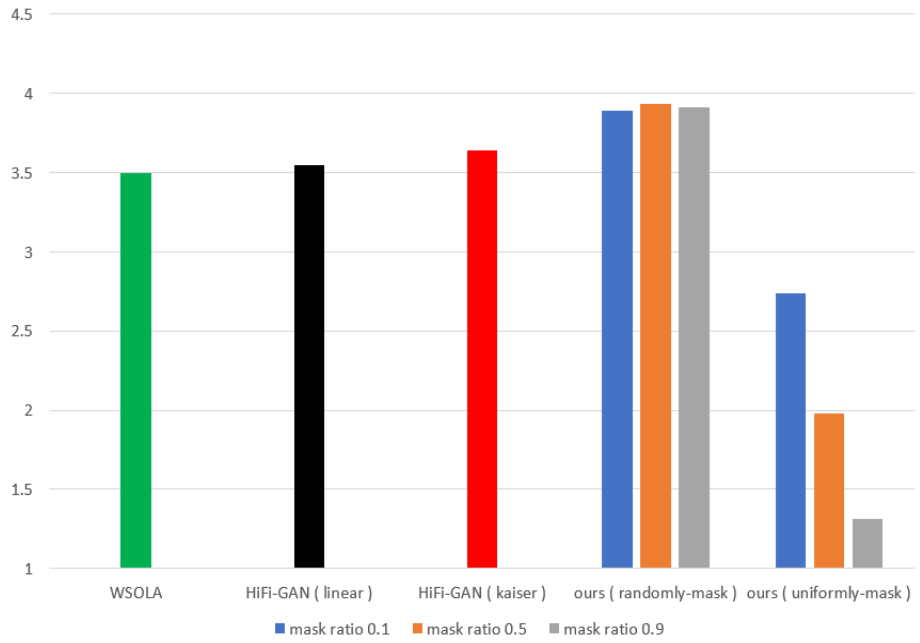


Fig. 4.7: The MOS of speech lengthened by WSOLA and the proposed method. The speech is lengthened to 4 / 3 of the original time.

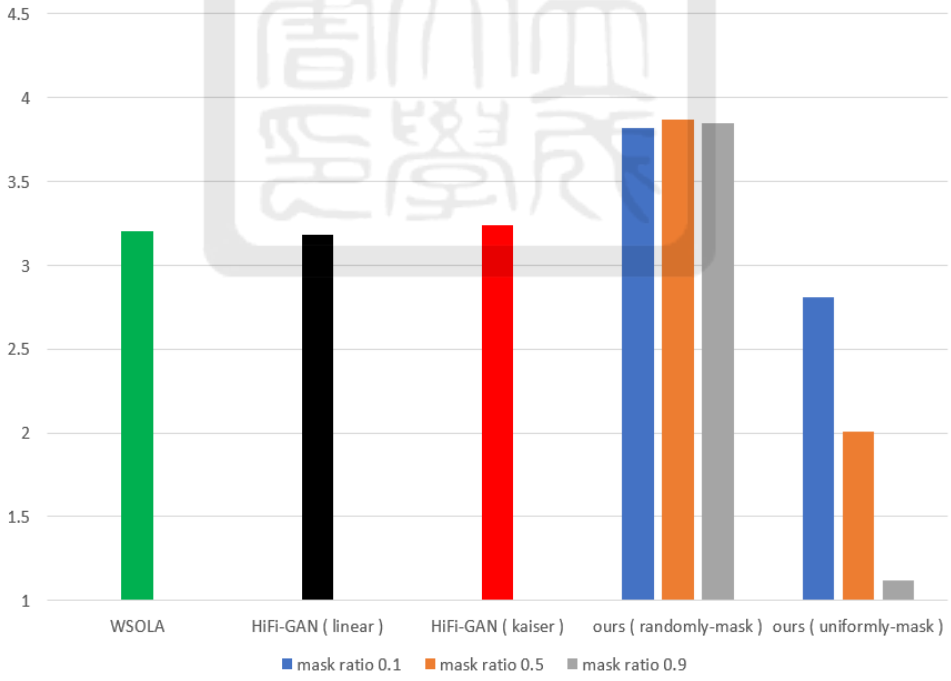


Fig. 4.8: The MOS of speech lengthened by WSOLA and the proposed method. The speech is lengthened to 3 / 2 of the original time.

4.2.3 Audio Signals with Duration Partially Modified

A speech signal with a natural prosody is usually spoken at different speed at different part of the sentence. Since the main purpose of the proposed aims at providing a speech with natural prosody, we will evaluate the naturalness of a speech signal with its duration modified only at some certain region other than totally lengthen or shorten. In this experiment, we recorded speeches from real human. For comparison, we synthesize the same sentence from a TTS model and use our system to lengthen or shorten the duration at given region to mimic the prosody of real human speeches. Experimental results can be seen in fig. 4.9.

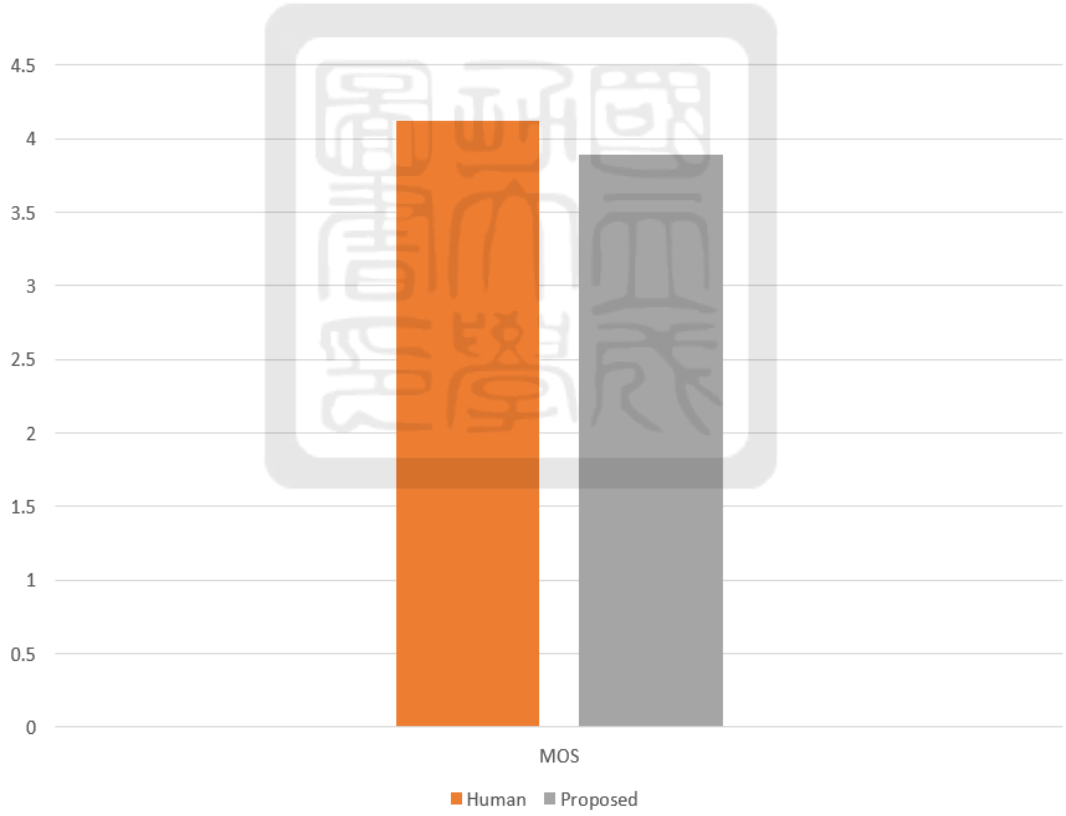


Fig. 4.9: The MOS of speeches by real human and the proposed system.

4.2.4 The Vocoders

To evaluate the effect that different vocoders have on the proposed system, in addition to the Voc-GAN, we adopt another non-neural vocoder in our proposed system in this experiment. The state-of-the-art non-neural vocoder in recent decades is the Griffin-Lim algorithm [38].

In this experiment, the Griffin-Lim algorithm and the VOC-Gan are used in both lengthening and shortening cases of our proposed system. We use the ATM-Net trained with randomly-masked strategy. The masking ratio is set to 0.5. The MOS of each combination is listed in fig. 4.9.

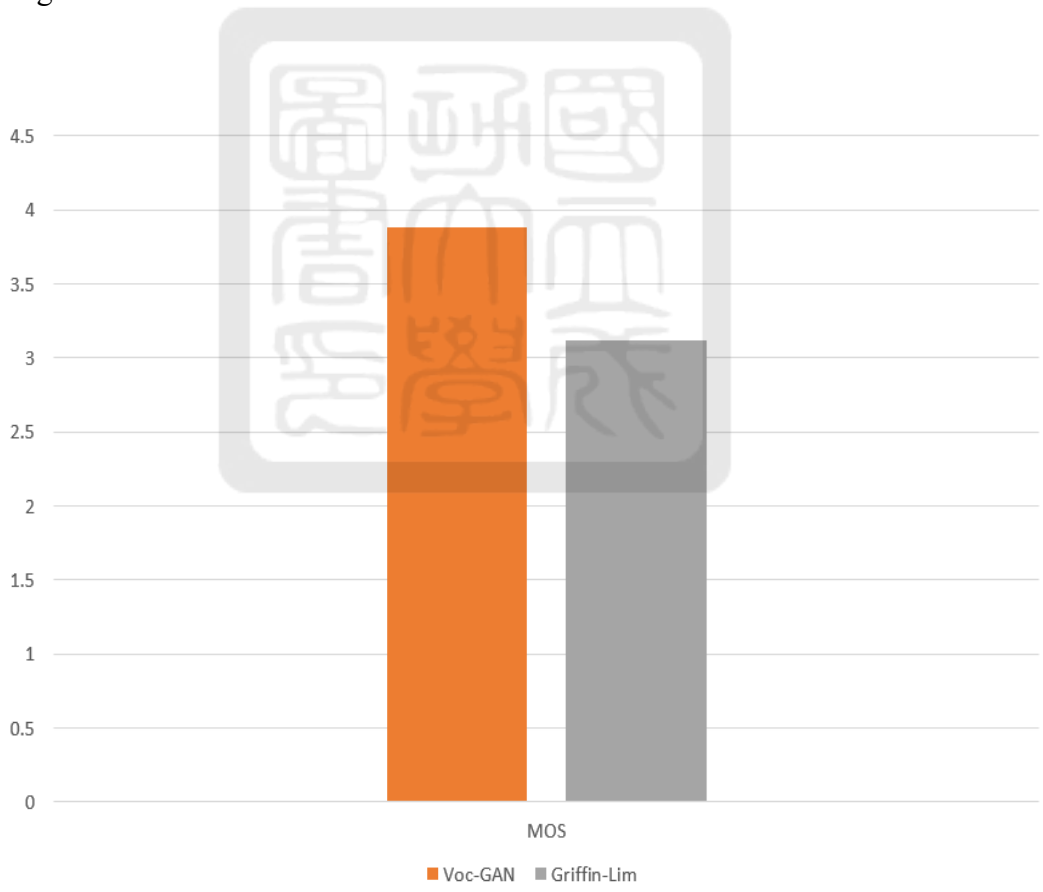


Fig. 4.10: The MOS of speeches generated by different vocoders.

4.3 Discussions

4.3.1 Discussions on the masking strategies

In the first and the second experiments, it could be seen that the proposed system performs slightly better comparing with the other two algorithms in shortening the duration of audio signals. Moreover, there are no significant differences between the MOS of different masking strategies or masking ratio. Furthermore, the Cohen's Kappa Coefficient of the MOS is 0.897, showing that all the listeners have reached a high agreement.

On the other hand, in lengthening audio signals, the proposed system trained with random-masked strategy out-performs the other two algorithms especially when the audio is stretched longer. In the comparison between two training methods, we think the reason why the system trained with random-masked strategy performs significantly better lies in that randomly generated mask would gather together to cover a diversity of mask lengths, which resembles different lengths of dummy frames inserted. Therefore, as the length of the audio signal are stretched longer, the performance of the system trained with uniformly generated masks would degrade significantly. We have analyzed the distribution of mask lengths of the randomly generated masks. The result is shown in table 4.3 to table 4.8. It could be seen that the majority of mask lengths lies between 1 to 4. However, as the masking ratio grows, the longest length of mask length could reach up to 28. Therefore, the system trained with this stretching method would generally perform better with diverse lengthening ratio.

Table 4.3: Statistics of mask length of dataset F01 with mask ratio 0.1

Mask length	1	2	3	4
Count	182817	17949	2509	331
Mask length	5	6	7	8
Count	33	11	1	0

Table 4.4: Statistics of mask length of dataset F01 with mask ratio 0.5

Mask length	1	2	3	4
Count	339233	134236	52969	21245
Mask length	5	6	7	8
Count	8415	3276	1259	489
Mask length	9	10	11	12
Count	196	67	34	15
Mask length	13	14	15	16
Count	0	4	0	0

Table 4.5: Statistics of mask length of dataset F01 with mask ratio 0.9

Mask length	1	2	3	4
Count	229123	136566	81208	48765
Mask length	5	6	7	8
Count	28778	17051	10109	6018
Mask length	9	10	11	12
Count	3557	2078	1231	748
Mask length	13	14	15	16
Count	407	264	162	84
Mask length	17	18	19	20
Count	47	43	25	15
Mask length	21	22	23	24
Count	10	4	4	2
Mask length	25	26	27	28
Count	0	0	0	1

Table 4.6: Statistics of mask length of dataset M01 with mask ratio 0.1

Mask length	1	2	3	4
Count	182820	17884	2601	313
Mask length	5	6	7	8
Count	39	3	1	0

Table 4.7: Statistics of mask length of dataset M01 with mask ratio 0.5

Mask length	1	2	3	4
Count	339013	134324	52956	21061
Mask length	5	6	7	8
Count	8503	3250	1359	547
Mask length	9	10	11	12
Count	187	73	34	14
Mask length	13	14	15	16
Count	4	3	0	0

Table 4.8: Statistics of mask length of dataset M01 with mask ratio 0.9

Mask length	1	2	3	4
Count	229531	136248	81371	48139
Mask length	5	6	7	8
Count	28884	17301	10218	6011
Mask length	9	10	11	12
Count	3525	2144	1220	759
Mask length	13	14	15	16
Count	436	279	158	95
Mask length	17	18	19	20
Count	40	36	23	13
Mask length	21	22	23	24
Count	10	5	4	2
Mask length	25	26	27	28
Count	2	0	1	1

4.3.2 A Brief History of the ATM-Net

As a matter of fact, in addition to the ATM-Net and the two-staged training method, we have tried several other methods to lengthen the speech. More specifically, we have tried several networks with different architecture to fill in the dummy frames inserted to lengthen the speech. Therefore, we will briefly discuss the history of how we come up with the ATM-Net and the two-staged training method.

First, we tried to use the same network architecture as that of the SSRN to fill in the dummy frames. The SSRN is trained using the mel spectrogram reduced to one forth in time scale as input, and the full mel spectrogram as output. At the inference stage, the SSRN is then fed with a mel spectrogram with dummy frames. However, the SSRN trained in this manner have not seen any dummy frame inserted between the original frames. Therefore, the resulting output mel spectrogram appears to be almost blank even if only a few dummy frames are inserted.

The second attempt we have tried is to use an encoder-decoder pair to synthesize the dummy frames in a recursive manner. In the training stage, the input mel spectrogram is first cut into several segments with a fix-sized window. The frame at the middle of each segment is masked. The encoder-decoder pair is trained to re-synthesize the masked frame with the previous and the following frames in the segment. In the inference stage, the dummy frames inserted are filled in by the encoder-decoder pair by taking the previous and the following original frames into account. The training and inference procedure are shown in fig. 4.10 and 4.11 respectively. This method, despite the fact that it could successfully generate the content in the inserted dummy frame, has a major drawback that there will be an obvious discontinuity between the original frame and the synthesized frame. As shown in fig. 4.12.

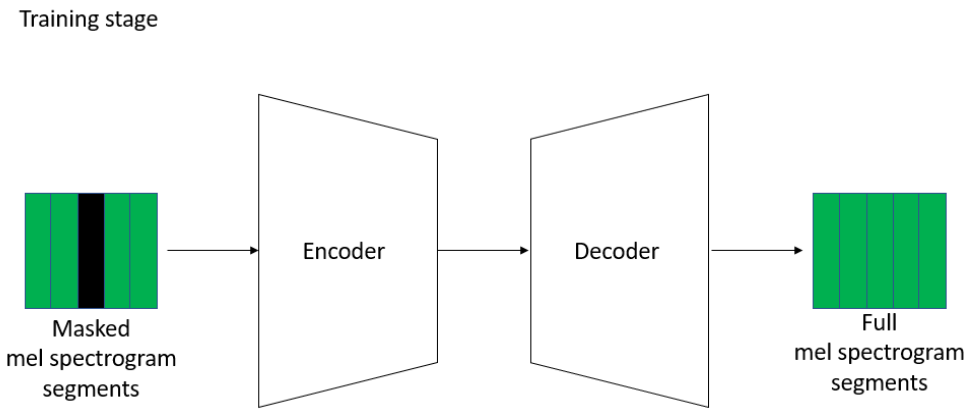


Fig. 4.11: The training process of the encoder-decoder model. The model is trained to fill in the mask at the middle of the mel spectrogram segment.

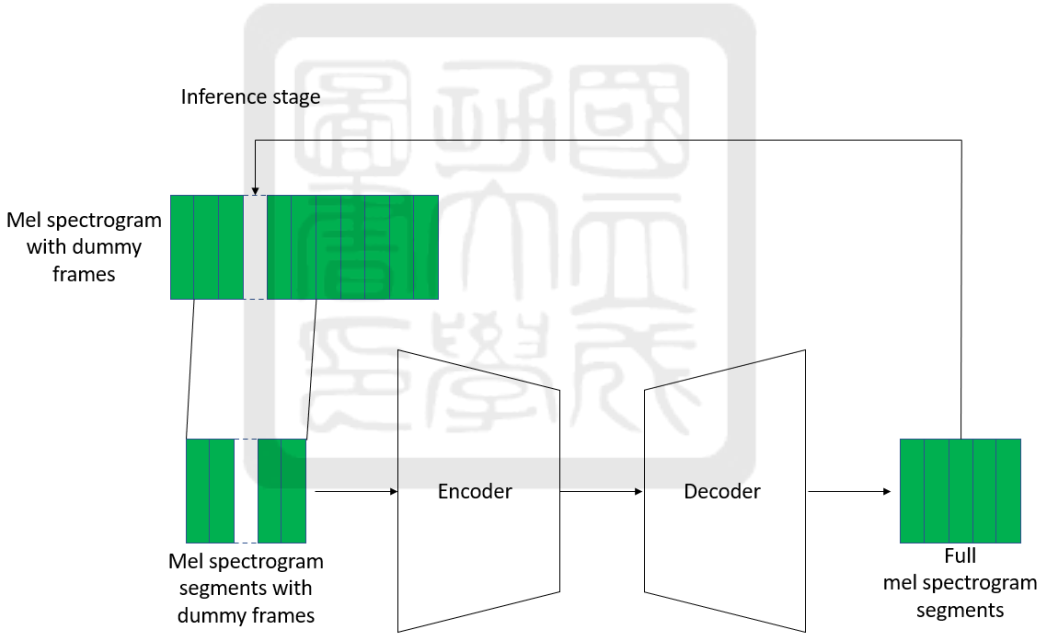
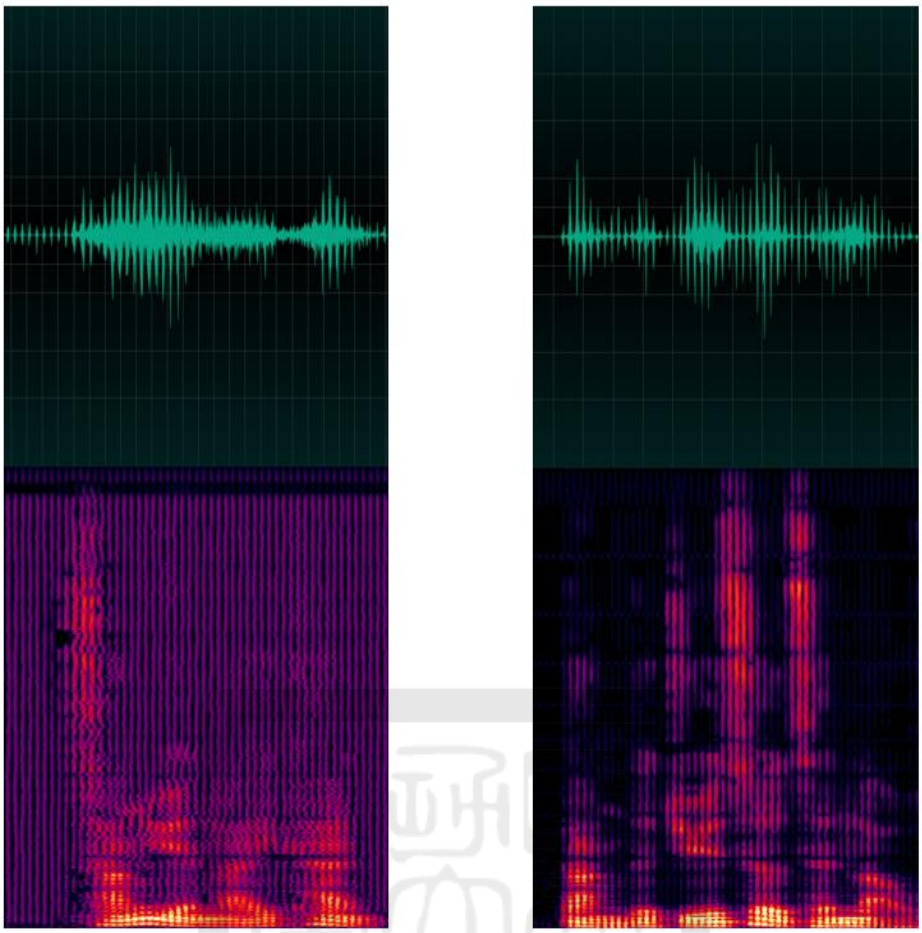


Fig. 4.12: The inference method of the encoder-decoder model. Given a mel spectrogram with dummy frames. The encoder-decoder model recursively extracts segments containing dummy frames and fills in the dummy frame using the encoder-decoder model.

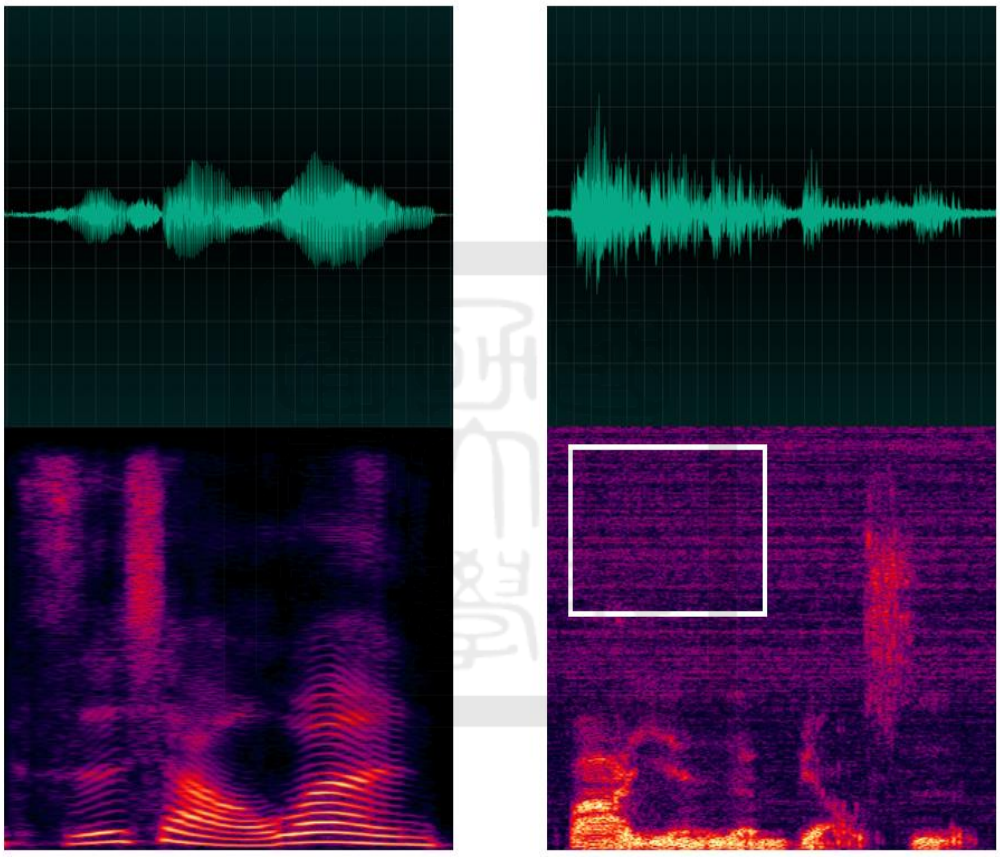


(a) (b)

Fig. 4.13: The inference samples of the encoder-decoder pair model. (a) The length of a segment equals to 50 frames. (b) The length of a segment equals to 150 frames. It appears to be obvious discontinuity between generated frames regardless of the difference on the length of a segment.

As for the third method, we start to use the architecture of the ATM-Net proposed in this thesis and train the ATM-Net from scratch with masked mel spectrogram. This method successfully repairs the discontinuity between the original frame and the synthesized frame, and thus gives us the first success to lengthen the input speech at desired position. However, there are still some hearable defects in the synthesized speech which downgrade the quality of the speech. As shown in fig. 4.13. In order to further improve the quality of the synthesized

speech, we come up with the two-staged training method. The intuition behind the two-staged method lies in that it treats the ATM-Net as a pure encoder-decoder pair of mel spectrogram in the first stage of training. Afterwards, in the second stage, the ATM-Net is the fine-tuned for the generation of the contents to fill in the dummy frames inserted between the original frames. Up to this point, the ATM-Net and the two-staged training method are finally proposed.



(a) (b)

Fig. 4.14: Effects of the two-staged training method. (a) The inference sample of the ATM-Net trained with the two-staged training method. (b) The inference sample of the ATM-Net trained from scratch. It could be seen that there's an obvious noise in the inference sample trained from scratch, as shown in the white box.

Chapter 5 Conclusion and Future Works

5.1 The Conclusion

Synthesizing speech from a TTS model has received great attention in the past few decades. In addition to synthesizing intelligible speech, more and more efforts have been spent on modeling the duration of the synthesized speech to control the prosody of speech. One common way of modeling the duration of the synthesized speech is to use a non-autoregressive TTS model. Despite the success that non-autoregressive TTS models have reached in synthesizing high-quality speeches as well as having the controllability of the duration of the synthesized speech, the training of a non-autoregressive TTS model requires a huge amount of training data and takes large computing power, thus making it difficult to training a non-autoregressive TTS model. Therefore, in this thesis, we proposed a system that could adjust the duration of a speech that is synthesized from a TTS model, no matter an autoregressive or a non-autoregressive TTS model. The system takes the synthesized speech from an TTS model and the corresponding transcript as input, then outputs the speech with its duration modified corresponding to the desired stretching ratio. The system consists of a forced aligner, a duration modifier, an ATM-Net and a vocoder.

The architecture of the ATM-Net is adopted from the SSRN with three main modifications. First, the deconvolutional layers are removed since the stretching ratio of different parts of the speech can be different. Second, the highway convolutional blocks are replaced by residual blocks for deeper network architecture. Third, the output dimension of the ATM-Net is changed to 80 to fit the dimension of a mel spectrogram, which is used for a vocoder to synthesize a speech signal. Moreover, we proposed a two-staged training method for the training of the ATM-Net. At the first stage of the training process, the ATM-Net is trained as an encoder / decoder pair that synthesize exact the same mel spectrogram

as the input mel spectrogram. At stage 2, the ATM-Net is further finetuned to fill in proper audio contents in the dummy frames inserted by the duration modifier and meanwhile smoothen the discontinuity between frames introduced by inserting and removing frames in a mel spectrogram. At this stage of training, input mel spectrograms are masked and the ATM-Net is supposed to learn to fill in proper audio contents in the masked frames. Two masking strategies, a randomly distributed masking strategy or a uniformly distributed masking strategy, are proposed. Experimental results show that our system could adjust the duration of speech and the quality of the output speech is better than the output of the WSOLA algorithm and the speaking-rate-changing HiFi-GAN. Furthermore, we have found that in the 2-staged training process, the randomly distributed masking strategy results in a better performance of the ATM-Net compared to the uniformly distributed masking strategy.

5.2 The Future Works

For future works, one possible way to improve the proposed system lies in the shortening of the input speech. In order to shorten the input speech, the duration modifier in the proposed system removes a few frames at the desired region of the input speech. However, when the number of frames to be removed are larger than that of the frames at the desired region, either a word or a phoneme, the whole word or phoneme will be completely removed and will disappear in the output speech. To solve this problem, further efforts should be spent on the strategies of shortening the duration of speech.

Reference

[1] Wang, Y., Skerry-Ryan, R. J., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q., Agionyrgiannakis, Y., Clark, R., Saurous, R. A. (2017). Tacotron: Towards end-to-end speech synthesis. arXiv preprint arXiv:1703.10135.

[2] Sotelo, J., Mehri, S., Kumar, K., Santos, J. F., Kastner, K., Courville, A., & Bengio, Y. (2017). Char2wav: End-to-end speech synthesis.

[3] Ping, W., Peng, K., Gibiansky, A., Arik, S. O., Kannan, A., Narang, S., Raiman, J., Miller, J. (2017). Deep voice 3: Scaling text-to-speech with convolutional sequence learning. arXiv preprint arXiv:1710.07654.

[4] Li, N., Liu, S., Liu, Y., Zhao, S., & Liu, M. (2019, July). Neural speech synthesis with transformer network. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 6706-6713).

[5] Ren, Y., Ruan, Y., Tan, X., Qin, T., Zhao, S., Zhao, Z., & Liu, T. Y. (2019). Fastspeech: Fast, robust and controllable text to speech. Advances in neural information processing systems, 32.

[6] Ren, Y., Hu, C., Tan, X., Qin, T., Zhao, S., Zhao, Z., & Liu, T. Y. (2020). Fastspeech 2: Fast and high-quality end-to-end text to speech. arXiv preprint arXiv:2006.04558.

[7] Elias, I., Zen, H., Shen, J., Zhang, Y., Jia, Y., Weiss, R. J., & Wu, Y. (2021, June). Parallel tacotron: Non-autoregressive and controllable tts. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5709-5713). IEEE.

[8] Yu, C., Lu, H., Hu, N., Yu, M., Weng, C., Xu, K., Liu, P., Tou, D., Kang, S., Lei, G., Su, D., Yu, D. (2019). Durian: Duration informed attention network for multimodal synthesis. arXiv preprint arXiv:1909.01700.

[9] Łańcucki, A. (2021, June). Fastpitch: Parallel text-to-speech with pitch prediction. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6588-6592). IEEE.

[10] Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerry-Ryan, R. J., Saurous, R. A., Agiomyrgiannakis, Y., Wu, Y. (2018, April). Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 4779-4783). IEEE.

[11] Lim, D., Jang, W., Park, H., Kim, B., & Yoon, J. (2020). Jdi-t: Jointly trained duration informed transformer for text-to-speech without explicit alignment. arXiv preprint arXiv:2005.07799.

[12] Beliaev, S., Rebryk, Y., & Ginsburg, B. (2020). TalkNet: Fully-convolutional non-autoregressive speech synthesis model. arXiv preprint arXiv:2005.05514.

[13] Kriman, S., Beliaev, S., Ginsburg, B., Huang, J., Kuchaiev, O., Lavrukhin, V., Leary, R., Li, J., Zhang, Y. (2020, May). Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6124-6128). IEEE.

[14] Zeng, Z., Wang, J., Cheng, N., Xia, T., & Xiao, J. (2020, May). Alignтts: Efficient feed-forward text-to-speech system without explicit alignment. In ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 6714-6718). IEEE.

[15] McAuliffe, M., Socolof, M., Mihuc, S., Wagner, M., & Sonderegger, M. (2017, August). Montreal forced aligner: Trainable text-speech alignment using kaldi. In Interspeech (Vol. 2017, pp. 498-502).

[16] Shen, J., Jia, Y., Chrzanowski, M., Zhang, Y., Elias, I., Zen, H., & Wu, Y. (2020). Non-attentive tacotron: Robust and controllable neural tts synthesis including unsupervised duration modeling. arXiv preprint arXiv:2010.04301.

[17] Elias, I., Zen, H., Shen, J., Zhang, Y., Jia, Y., Skerry-Ryan, R. J., & Wu, Y. (2021). Parallel Tacotron 2: A non-autoregressive neural TTS model with differentiable duration modeling. arXiv preprint arXiv:2103.14574.

[18] Abbas, A., Merritt, T., Moinet, A., Karlapati, S., Muszynska, E., Slangen, S., Gatti, E., Drugman, T. (2022). Expressive, variable, and controllable duration modelling in TTS. arXiv preprint arXiv:2206.14165.

[19] Allen, J. B., & Rabiner, L. R. (1977). A unified approach to short-time Fourier analysis and synthesis. Proceedings of the IEEE, 65(11), 1558-1564.

[20] Roucos, S., & Wilgus, A. (1985, April). High quality time-scale modification for speech. In ICASSP'85. IEEE International Conference on Acoustics, Speech, and Signal Processing (Vol. 10, pp. 493-496). IEEE.

[21] Rudresh, S., Vasisht, A., Vijayan, K., & Seelamantula, C. S. (2018). Epoch-synchronous overlap-add (ESOLA) for time-and pitch-scale modification of speech signals. arXiv preprint arXiv:1801.06492.

[22] Laroche, J. (1993, October). Autocorrelation method for high-quality time/pitch-scaling. In Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (pp. 131-134). IEEE.

[23] Lawlor, B., & Fagan, A. D. (1999). A novel high quality efficient algorithm for time-scale modification of speech.

[24] Wong, P. H., & Au, O. C. (2003). Fast SOLA-based time scale modification using envelope matching. Journal of VLSI signal processing systems for signal, image and video technology, 35, 75-90.

[25] Wong, P. H., & Au, O. C. (2002, May). Fast SOLA-based time scale modification using modified envelope matching. In 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (Vol. 3, pp. III-3188). IEEE.

[26] Dorran, D., Lawlor, R., & Coyle, E. (2003, April). High quality time-scale modification of speech using a peak alignment overlap-add algorithm (PAOLA). In 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). (Vol. 1, pp. I-I). IEEE.

[27] Xin, D., Takamichi, S., Okamoto, T., Kawai, H., & Saruwatari, H. (2022). Speaking-Rate-Controllable HiFi-GAN Using Feature Interpolation. arXiv preprint arXiv:2204.10561.

[28] Kong, J., Kim, J., & Bae, J. (2020). Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. Advances in Neural Information Processing Systems, 33, 17022-17033.

[29] Viikki, O., & Laurila, K. (1998). Cepstral domain segmental feature vector normalization for noise robust speech recognition. Speech Communication, 25(1-3), 133-147.

[30] Povey, D., & Saon, G. (2006, September). Feature and model space speaker adaptation with full covariance gaussians. In Interspeech (pp. 1145-1148).

[31] McAuliffe, M., Socolof, M., Mihuc, S., Wagner, M., & Sonderegger, M. "*MFA Pretrained Mandarin Models in International Phonetic Alphabet*", https://montreal-forced-aligner.readthedocs.io/en/latest/user_guide/models/index.html

[32] Tachibana, H., Uenoyama, K., & Aihara, S. (2018, April). Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention. In 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 4784-4788). IEEE.

[33] Yang, J., Lee, J., Kim, Y., Cho, H., & Kim, I. (2020). VocGAN: A high-fidelity real-time vocoder with a hierarchically-nested adversarial network. arXiv preprint arXiv:2007.15256.

[34] Rix, A. W., Beerends, J. G., Hollier, M. P., & Hekstra, A. P. (2001, May). Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In 2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221) (Vol. 2, pp. 749-752). IEEE.

[35] FFmpeg, http://ffmpeg.org

[36] Verhelst, W., & Roelands, M. (1993, April). An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech. In 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing (Vol. 2, pp. 554-557). IEEE.

[37] ffmpeg atempo API, https://ffmpeg.org/ffmpeg-filters.html#atempo

[38] Griffin, D., & Lim, J. (1984). Signal estimation from modified short-time Fourier transform. IEEE Transactions on acoustics, speech, and signal processing, 32(2), 236-243.

[39] I. Rosenfelder, J. Fruehwald, K. Evanini, and J. Yuan, "FAVE (Forced Alignment and Vowel Extraction) Program Suite [Computer program]," 2011, available at http://fave.ling.upenn.edu.

[40] Brugnara, F., Falavigna, D., & Omologo, M. (1993). Automatic segmentation and labeling of speech based on Hidden Markov Models. Speech Communication, 12(4), 357-370.

[41] Forney, G. D. (1973). The viterbi algorithm. Proceedings of the IEEE, 61(3), 268-278.

[42] Malfrere, F., & Dutoit, T. (1997). High-quality speech synthesis for phonetic speech segmentation. In Fifth European Conference on Speech Communication and Technology.

[43] van Santen, J. P., & Sproat, R. (1999, September). High-accuracy automatic segmentation. In EUROSPEECH.

[44] Katsamanis, A., Black, M., Georgiou, P. G., Goldstein, L., & Narayanan, S. (2011, January). SailAlign: Robust long speech-text alignment. In Proc. of workshop on new tools and methods for very-large scale phonetics research (Vol. 1).

[45] Morise, M., Yokomori, F., & Ozawa, K. (2016). WORLD: a vocoder-based high-quality speech synthesis system for real-time applications. IEICE TRANSACTIONS on Information and Systems, 99(7), 1877-1884.

[46] Morise, M., Kawahara, H., & Katayose, H. (2009, February). Fast and reliable F0 estimation method based on the period extraction of vocal fold vibration of singing voice and speech. In Audio Engineering Society Conference: 35th International Conference: Audio for Games. Audio Engineering Society.

[47] Morise, M. (2015). CheapTrick, a spectral envelope estimator for high-quality speech synthesis. Speech Communication, 67, 1-7.

[48] Morise, M. (2012). Platinum: A method to extract excitation signals for voice synthesis system. Acoustical Science and Technology, 33(2), 123-125.

[49] Prenger, R., Valle, R., & Catanzaro, B. (2019, May). Waveglow: A flow-based generative network for speech synthesis. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 3617-3621). IEEE.

[50] Kingma, D. P., & Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. Advances in neural information processing systems, 31.

[51] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems, 27.

[52] Yamamoto, R., Song, E., & Kim, J. M. (2020, May). Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6199-6203). IEEE.

[53] Kumar, K., Kumar, R., De Boissiere, T., Gestin, L., Teoh, W. Z., Sotelo, J., De Brebisson, A., Bengio, Y., Courville, A. C. (2019). Melgan: Generative adversarial networks for conditional waveform synthesis. Advances in neural information processing systems, 32.

[54] Stevens, S. S., Volkmann, J., & Newman, E. B. (1937). A scale for the measurement of the psychological magnitude pitch. The journal of the acoustical society of america, 8(3), 185-190.

[55] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[56] The VocGAN, https://github.com/rishikksh20/VocGAN.

[57] Keith Ito and Linda Johnson, "The LJ Speech Dataset", https://keithito.com/LJ-Speech-Dataset/, 2017.

[58] pysox tempo, https://pysox.readthedocs.io/en/latest/api.html

[59] LeCun, Y., & Bengio, Y. (1998). The handbook of brain theory and neural networks.

[60] Srivastava, R. K., Greff, K., & Schmidhuber, J. (2015). Training very deep networks. Advances in neural information processing systems, 28.

[61] McHugh, M. L. (2012). Interrater reliability: the kappa statistic. Biochemia medica, 22(3), 276-282.