공학박사 학위논문

조건부 자기회귀형 인공신경망을 이용한 제어 가능한 가창 음성 합성

Controllable Singing Voice Synthesis using Conditional Autoregressive Neural Network

2022 년 8 월

서울대학교 융합과학기술대학원

지능정보융합학과 지능정보융합학전공

이 주 헌

공학박사 학위논문

조건부 자기회귀형 인공신경망을 이용한 제어 가능한 가창 음성 합성

Controllable Singing Voice Synthesis using Conditional Autoregressive Neural Network

2022 년 8 월

서울대학교 융합과학기술대학원

지능정보융합학과 지능정보융합학전공

이 주 헌

조건부 자기회귀형 인공신경망을 이용한 제어 가능한 가창 음성 합성

Controllable Singing Voice Synthesis using Conditional Autoregressive Neural Network

지도교수 이 교 구

이 논문을 공학박사 학위논문으로 제출함

2022 년 8 월

서울대학교 융합과학기술대학원

지능정보융합학과 지능정보융합학전공

이 주 헌

이주헌의 공학박사 학위논문을 인준함

2022 년 8 월

위 원 장	이 원 종	(인)
부위원장	이 교 구	(인)
위 원	곽 노 준	(인)
위 원	서 봉 원	(인)
위 원	남 주 한	(인)

Abstract

Singing voice synthesis aims at synthesizing a natural singing voice from given input information. A successful singing synthesis system is important not only because it can significantly reduce the cost of the music production process, but also because it helps to more easily and conveniently reflect the creator's intentions. However, there are three challenging problems in designing such a system - 1) It should be possible to independently control the various elements that make up the singing. 2) It must be possible to generate high-quality sound sources, 3) It is difficult to secure sufficient training data. To deal with this problem, we first paid attention to the source-filter theory, which is a representative speech production modeling technique. We tried to secure training data efficiency and controllability at the same time by modeling a singing voice as a convolution of the source, which is pitch information, and filter, which is the pronunciation information, and designing a structure that can model each independently. In addition, we used a conditional autoregressive model-based deep neural network to effectively model sequential data in a situation where conditional inputs such as pronunciation, pitch, and speaker are given. In order for the entire framework to generate a high-quality sound source with a distribution more similar to that of a real singing voice, the adversarial training technique was applied to the training process. Finally, we applied a self-supervised style modeling technique to model detailed unlabeled musical expressions. We confirmed that the proposed model can flexibly control various elements such as pronunciation, pitch, timbre, singing style, and musical expression, while synthesizing high-quality singing that is difficult to distinguish from ground truth singing. Furthermore, we proposed a generation and modification framework that considers the situation applied to the actual music production process, and confirmed that it is possible to apply it to expand the limits of the creator's imagination, such as new voice design and cross-generation.

Keywords: singing voice synthesis, source-filter theory, conditional autoregressive neural network, singer identity, musical expression

Student Number: 2018-23798

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Singing voice synthesis (SVS) is a task aimed at synthesizing a natural singing voice from given input information. The elements that can be used as input information of the SVS are composed of the essential elements of singing, such as musical score, lyrics, musical expression, singer's voice, and singing style. The musical score contains the start time, duration, and pitch information for each musical note. The lyrics contain information on how to pronounce each musical note. The musical expression symbols that mean the expression of each note's stress, speed, vibrato, etc., may also be used as input. Finally, it should be possible to input information related to the identity of the singer, which determines with what voice and how to sing the song. In summary, the SVS system needs to understand the various elements that make up a song and generate a natural singing voice as if a human-based sang it on this.

Similar to the development of electronic music, which allows music producers to create music using a variety of sounds without directly playing an instrument, singing voice synthesis can be used to lower the production cost for singing in the music production system. In general, producing music that includes vocals involves several steps, such as finding a singer who suits the song, hiring a singer, and then spending time and money in the studio to record several times to get the desired result. If producers use the SVS system, they can find a singer with the desired timbre from a predefined singer database and quickly create a song reflecting various expressions. In addition, various musical attempts that were not possible in the existing method can be made possible. For example, we can create a new timbre by combining two different singers' timbre, creating a song in multiple languages with one singer's voice, or creating a new identity that combines the timbre and singing style of two singers independently. With the SVS technology, faster prototyping in music production is possible. Combining various attempts that were previously impossible can easily reflect the creative imagination of the producer and make richer musical expression possible.

To this end, the requirements for a successful SVS system are as follows. First, it should be possible to independently control each of the various elements that make up singing. Unlike general speech, singing has a set beat and rhythm, so we need a system that can control the pronunciation and pitch to be uttered at a given time. In addition, it should be able to control what voice and how they sing the song. For this, a system that can understand and analyze the identity of a given singer and reflect it in the generation process is needed. Furthermore, to create a more expressive singing, a method that can input and control various musical expressions is also required. It can be expressed strongly or weakly, fast or slow, monotonously or dynamically, even for the same musical note. In order to more directly reflect the intention of the creator, it is necessary to be able to control these musical expressions properly.

Second, it is necessary to ensure the quality that the output generated through the SVS system can be used in the actual music industry. In general, sound sources handled by the music industry are recorded in a controlled environment with high quality. In contrast to most speech synthesis research focusing on modeling sound sources with sampling rates of 22050 kHz or lower, singing synthesis research should be able to generate high-quality sound sources of 44100 kHz or higher.

Third, a convenient production interface should be provided to the user. A typical song recording process is based on numerous trials, errors, and revisions. In order to achieve this process through the singing voice synthesis system, it is necessary to have the ability to modify and reproduce the result obtained as desired, beyond simply generating high-quality singing.

In this thesis, we propose a method to construct an expressive and controllable singing voice synthesis system that can be used in music production industry. We proposed a novel architecture that independently models pitch and pronunciation to construct an SVS system that can effectively model various input elements. In addition, it has been expanded with a novel multi-singer SVS system that separates and controls the singer's singing style and voice timbre. By proposing a local style token module that extracts expressive elements from the input musical score itself based on a self-supervised manner, expressive power and controllability were improved. A dual-path pitch encoder was introduced to provide a convenient methodology for re-modifying the generated signal.

1.2 Problems in singing voice synthesis

In order to design a high-quality singing voice synthesis system that can be used as a creative tool, there are various challenging problems. In this chapter, we divide the problems into three categories - datasets, controllability, and usability - and describe them.

Dataset A singing voice synthesis system is designed using a clean singing signal recorded in a controlled environment and corresponding score music information. In the conventional method, the concatenative method, after collecting songs sung by a singer in various pitches and pronunciations in advance, selects appropriate pieces according to the input musical score query, combines them, and attaches them. A deep neural network-based singing voice synthesis system, which has been receiving a lot of attention recently, uses a method to design a neural network that directly maps a target waveform from condition input information and train it based on a supervised learning method. In both cases, a high-quality singing voice data set is required in common.

However, it is not easy to collect a bunch of high-quality singing data in general. First, there is an issue related to the copyright of data. Since most singing data is created and published for commercial use, copyright is granted to those involved. Therefore, there may be various restrictions on the mass sharing with the public for research purposes. Second, it is not easy to obtain a separated signal where only a singing voice exists. As with the first issue, most songs are mixed and released together with the accompaniment music. For modeling the learning-based singing voice synthesis system, a clean sound source to which vocal effects are not applied and separated from the accompaniment is required, but these data are not well disclosed. Finally, there is a problem that a lot of time and money is consumed for proper annotation. In order to make a pair of sheet music and lyrics information corresponding to singing, the effort of a professional annotator is required. This is because the voice must be segmented into notes, and the pitch and pronunciation of each note must be labeled with a delicate temporal alignment.

For these reasons, it is challenging to secure data in singing voice synthesis research. Although a public dataset that has resolved the copyright issue has been released for use for research purposes [2, 3, 4, 5, 6], as shown in Table 1.1, it has different languages and annotation formats. Its size is far insufficient compared to the speech study area. Therefore, it is necessary to consider how to efficiently model the singing voice by using the given limited size of data set.

Controllability Various components make up a singing voice. Most fundamentally, there is information expressed in the musical score. With respect to one syllable, at what time and with what pitch and length to sing can be determined through the score. Also, what strength a note should be sung and how smooth the connection between two neighboring notes should also become factors that make up a song. As a more global attribute, there is also information about what kind of voice and singing style will be singing. The components constituting the singing voice may include notes, musical expression, the identity of a singer. In order to freely use the singing voice synthesis system in the way the creator wants, the possibility of manipulating the above elements is required.

Table 1.1 Description of representative datasets used in speech synthesis and singing synthesis research. The singing dataset is insufficient in terms of the number of speakers, the total length, and the unity of the annotation type.

Singing	duration (h)	# singer	sampling rate	annotation type
JVS music [4]	2.2	100	24.0 kHz	f0
OpenCpop [6]	5.2	1	44.1 kHz	note, pitch, phoneme
NUS [2]	1.8	12	44.1 kHz	phoneme
CSD [3]	3.6	1	44.1 kHz	note, pitch
Kiritan [5]	1.1	1	96.0 kHz	note, pitch
Speech	duration (h)	# speaker	sampling rate	annotation type
LibriTTS [7]	585.0	2456	24.0 kHz	text
VCTK [8]	44.0	109	96.0 kHz	text
HiFiTTS [9]	292.0	10	44.1 kHz	text

By understanding and interpreting the notes and various musical expressions in the score input by the user, and the identity of the singer specified by the user, the SVS system should be able to create a result that reflects these factors.

However, it is not easy to make all elements independently and operable. First, there is the problem of elements that cannot get labels. Unlike singer information and sheet music information, which can be labeled relatively quickly, complex musical expression consumes a lot of time and money to label. Second, there is a problem that a more efficient modeling technique is required because it is difficult to secure all combinations of each element in the training data in a state where the size of the dataset is limited even for cases where labels are possible, such as sheet music and singer information.

Therefore, in order to effectively control the elements that make up a song in a situation where the size of the dataset is not sufficient, a method of independently modeling each element is needed. Also, in order to control unlabeled information, we have to consider a self-supervised methodology.

Usability For the singing voice synthesis system to be used in the actual music production industry, usability is also an essential factor to be considered. The user should be able to quickly and easily create a song, and at the same time, if there is content that needs to be corrected in the generated result, it should be easily reflected and can be corrected in detail. However, it is challenging to build a system that can be used easily and can be finely modified. In general, singing synthesis generates acoustic signals, which are more complex information, from musical scores, which are relatively simple symbolic information. For ease of use, natural singing should be generated even when a structured signal quantized as an input is used. However, it is necessary to enable fine-level control of a specific area of the signal for more precise control. When usability is prioritized, it is difficult to make detailed corrections because the input signal must be designed in a simple form. Conversely, if a detailed input signal is designed to be used, the usability is reduced because the user has to manually input a complex signal each time it is used. Therefore, to design a singing voice synthesis system that satisfies both factors, it is necessary to consider factors such as the input method and correction method for generated results.

The following summarizes the challenging problems that exist in the design process of the singing voice synthesis system. First, we need to consider how to utilize a limited dataset effectively. Second, an effective modeling technique that can independently control various elements of singing should be considered. Third, the system should be designed to allow for detailed modifications while quickly and easily created. In this study, various attempts to solve these problems were introduced over three stages of the study, and the details are explained in the following section.

1.3 Task of interest

This section describes the task of interest, in which we proceeded step-by-step to design a high-quality singing synthesis system that can control various elements into three research topics. The first subsection includes the composition of the single-speaker singing voice synthesis system, which is the basis of the entire thesis and deals with the independent modeling of pronunciation and pitch based on source-filter theory. The second subsection introduces the definition of a singer's identity in addition to pronunciation and pitch. It deals with extending the existing system to a multi-singer SVS network. The third subsection describe how to improve the expressive power of the SVS system by modeling unlabeled expression elements other than sheet music and singer information in a self-supervised manner.

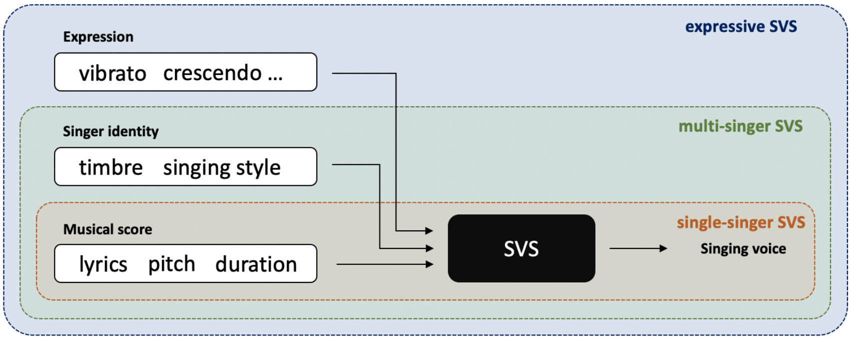


Fig. 1.1 Schematic diagram of the task of interest. A single-singer SVS that models the most basic information, a multi-singer SVS that can additionally control the identity information of a singer, and an expressive SVS that can additionally control various musical expressions.

1.3.1 Single-singer SVS

In the single-singer singing voice synthesis task, we focus on how to build an efficient system that can control and create elements of pitch, pronunciation, and length, which are the basics of singing voice synthesis. Based on the source-filter theory, which is a representative theory explaining the process of speech generation, effective modeling was carried out by designing two independent decoders that generate a source and a filter, respectively. In order to achieve more accurate pitch and pronunciation, a method for local conditioning of sheet music information at each stage of song generation was proposed to improve pronunciation and pitch accuracy. Adversarial training was introduced to the entire network training process to improve the production quality, and it was confirmed that the generation of higher-quality singing is possible.

1.3.2 Multi-singer SVS

In the multi-singer singing voice synthesis task, we focus on methods that can model the characteristics of various speakers based on a single speaker model. In this topic, we defined the singer's identity by dividing it into two independent elements: timbre and singing style. At this time, the timbre refers to the color of the voice that varies from person to person depending on the anatomical structure and size of the vocal cords. The singing style refers to an individual's method of interpreting and expressing given sheet music. In the multi-speaker singing synthesis task, we proposed a method to design an embedding vector that can efficiently contain the given singer information and a conditioning method reflected in the creation process. Through this, a system that can control the different characteristics of various people was built. In particular, a cross-generation method that intersects timbre and singing style was proposed, showing that the creative use of the singing synthesis system is possible.

1.3.3 Expressive SVS

The expressive singing voice synthesis task focuses on how to model unlabeled information. Various musical expressions other than sheet music and singers are challenging to secure because labeling costs. Therefore, in this study, the introduction of a local style token module that models various expression elements in a self-supervised learning-based method from the input of given sheet music contents was reviewed. In addition, a methodology to regenerate after modification based on the generated results was proposed to expand to a model with higher usability.

1.4 Contribution

The major contributions of the studies presented in this thesis can be summarized as follows:

1. **Singing voice synthesis network based on source-filter theory**: We proposed a singing voice synthesis system including two independent decoder structures based on the source-filter theory. By independently modeling pronunciation and pitch, it was confirmed that more efficient training was possible from a limited dataset. Furthermore, it was extended to a framework that can control singers' singing style and timbre.

2. **Self-supervised musical expression modeling**: We proposed a method to model information about unlabeled musical expressions from singing voice in a self-supervised manner. By introducing the concept of a local style token, it was possible to infer various musical expressions

on their own from the given sheet music information, and a method to modify and reproduce it from the generated results was proposed.

3. **High-quality, creative singing voice synthesis**: We achieved the high-quality level singing synthesis algorithm required in the actual music production process by introducing adversarial training and bandwidth extension vocoders. It was confirmed that a high-quality sound source of 44.1khz could be generated without losing controllability and usability of various elements, and various creative tasks such as cross-generation and identity fusion were possible that were not possible in existing recording methods.

Chapter 2

Background

This chapter describes the basic concepts and methodologies used in the thesis. We also include a review of papers related to this study throughout general speech synthesis research in this chapter. This chapter consists of four subsections: 1. singing voice, 2. source-filter theory, 3. autoregressive model, and 4. related works. First, we introduce the nature of the singing signal and the various elements of a singing voice signal. Next, source-filter theory, a representative model for the principle of human vocalization, will be explained. Thrid, the autoregressive modeling method that constitutes the basic model architecture constituting this paper will be described. Finally, Finally, we explain the flow of the speech synthesis research area and review related research papers to explain the significance of this study.

2.1 Singing voice

Singing voice refers to a voice signal expressed by the singer interpreting the given musical score and lyrics through his or her method. One singing voice signal can be expressed in various forms, and it is possible to express from the data signal itself, which is the lowest level description, to the score, which is the highest level description. A summary diagram of various expression forms for singing voice is shown in Figure 2.1. The singing voice synthesis system refers to generating a data signal, which is low-level information, from sheet music, which is high-level information. This subsection explains the basic concepts along with definitions of each step.

Waveform The waveform is a data format that records the amplitude of the sound recorded through the microphone over time. This is one of the most basic methods of representing sound, and a singing voice signal can also be expressed with this raw level information. The sampling rate of the waveform is determined according to how much data are measured per second. A high sampling rate is required to express sound with a wide frequency resolution. In general, it is common to use a sampling rate of 44.1 kHz or higher when making music. Therefore, the singing voice synthesis system must also be able to generate a sound source of 44.1kHz or higher.

Acoustic feature In order to generate a singing voice source with a sampling rate of 44.1kHz, it is necessary to predict about $180 * 44100 \ = 8M$ data points for 3 minutes long. The size of this information is significant, and it is challenging to model due to the characteristics of singing voice synthesis that

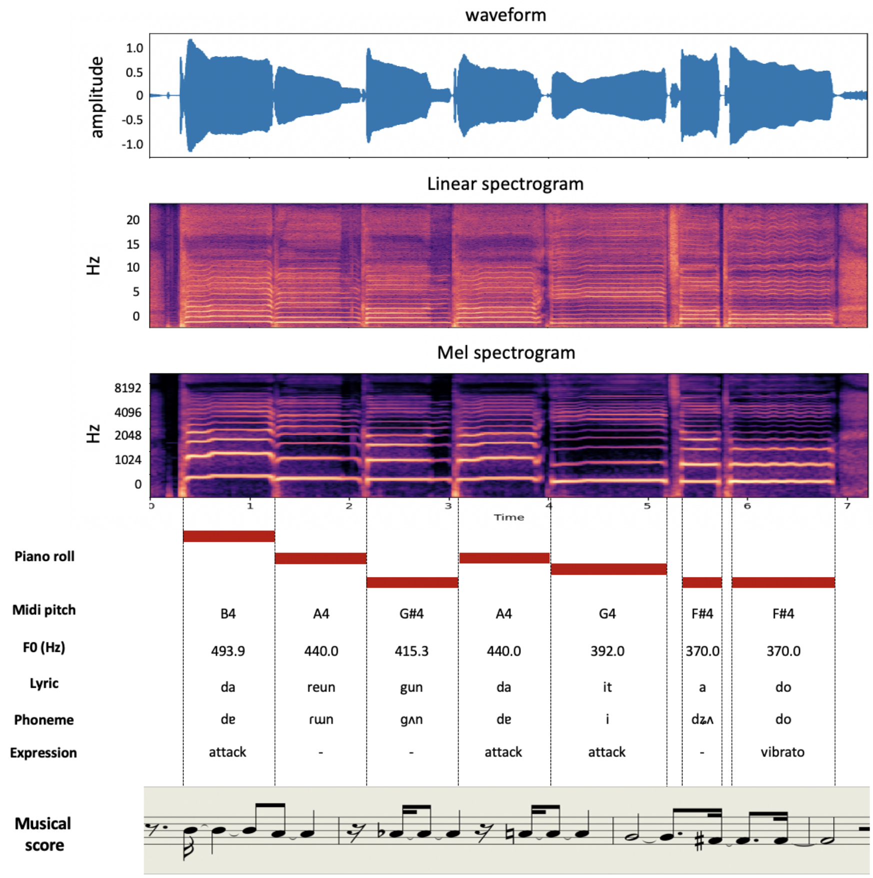


Fig. 2.1 A schematic diagram of various forms of representation for singing. A waveform, which is a raw-level representation, can be converted into an acoustic feature such as linear, mel spectrogram. The score, which is a high-level representation and symbolic information, contains various detailed elements such as pitch, pronunciation, and expression.

requires consideration of long-term dependency. Therefore, in general speech synthesis research, an acoustic feature of the corresponding waveform is generated instead of directly generating the raw waveform. After that, a vocoder that converts the acoustic feature into a waveform is additionally used.

Acoustic features should contain enough information to be converted into waveforms and, at the same time, be easy to model structurally. In general, the spectrogram, which is the result of applying the Fourier transform to the waveform, is used as an acoustic feature in the speech and singing synthesis research. This study uses the linear-spectrogram and the mel-spectrogram as acoustic features.

The linear spectrogram results from calculating the short-time Fourier transform on the waveform and is calculated as follows.

$$STFT\{x(n)\}[m, \omega] = \int_{-\infty}^{\infty} x[n]w[n-m]e^{-j\omega n} \tag{2.1}$$

where $w[]$ is the window function, and $STFT$ results from calculating discrete Fourier transform for a short window of a given waveform. As an acoustic feature, linear spectrogram indicates how much each frequency component is included in each short window and is suitable for containing characteristics such as pitch, length, and formant of a singing signal.

However, humans do not recognize frequency components based on a linear scale. People focus more on low-frequency components than high-frequency components and have high resolution. [10] suggests a frequency unit that listeners perceive as the same perceptual distance from each other. This is called the mel scale, and the result of converting the frequency axis of the linear spectrogram to the mel scale is called a mel spectrogram. The mel spectrogram is suitable for modeling as an acoustic feature because it is smaller in size and better reflects human perception than the linear spectrogram. Therefore, in this study, we used a method to predict a mel spectrogram from a musical score input and convert it into a linear spectrogram or waveform.

Musical score Another way of expressing a singing voice is a musical score. In general, a musical score is an expression form that contains the creator's intention about how to sing a given song. The musical score consists of musical notes and expressive elements. A note in a musical score of a singing voice contains information about a single syllable, including pitch, intensity, duration, lyrics, and timing. The creator expresses the song by the arrangement of notes, and the singer interprets the arrangement of notes and performs the song in their own way.

Another element included in the musical score is expressive symbols. Even notes with the same pronunciation, pitch, and length may have different expressions, such as singing gradually stronger (crescendo) or changing the pitch rapidly over time (vibrato). In addition, it is possible to express the overall slowness of singing over several note sequence areas (ritardando) or singing in short breaks (staccato). In this way, the techniques to be observed in expressing notes can be expressed as expression symbols in the musical score.

The singing voice synthesis system should be able to reflect the user's intention well. For this, it is necessary to design an appropriate input representation and to generate a result that reflects them well from the input. This study uses a note sequence containing start time, duration, pitch, and pronunciation information and additional expression information as input representations.

Singer identity The final factor that determines the singing considered in this paper is the singer's identity. Even when both the score and the expression symbol have been decided, the song can change depending on the style of the singer who interprets it. We define the identity of a singer by dividing it into timbre and singing style. Timbre refers to the color of the voice determined by the anatomical structure of the singer's vocal organs. The singing style refers to the singer's own method of interpreting the score as a song, and includes how to create a pitch contour from a given score, and how to allocate intensity over time.

In this study, we proposed a method to effectively model the identity of a singer while conducting a study to expand the single-singer singing voice synthesis system to the multi-singer singing voice synthesis system. In particular, a more creative music production process was made possible by suggesting a method of designing a new identity through combining a new voice and singing method.

2.2 Source-filter theory

This subsection describes the source-filter theory, which is the core basis of the singing voice synthesis system proposed in this paper. There are various input factors that determine singing, but the most important factors are pitch and pronunciation. According to the given musical score input, the singing voice synthesis system must determine which pronunciation and pitch to sing at every moment.

We paid attention to how humans use the vocal organs to generate singing.

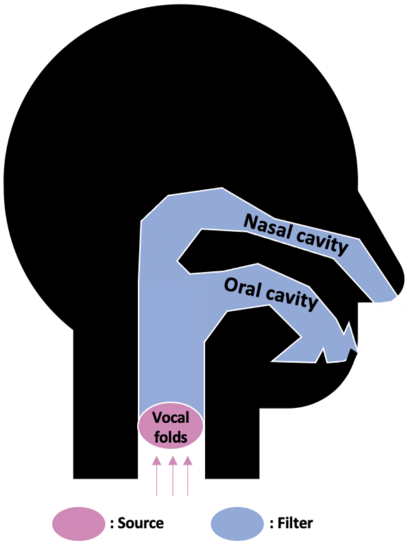


Fig. 2.2 Schematic diagram for source-filter theory. Air from the lungs vibrates the vocal folds, creating a source signal. The generated source signal passes through the nasal and oral cavity, is filtered, and then uttered as a voice.

The human voice is created by interacting with various organs in the path from the lungs to the mouth and nose. The flow of air by the pressure generated in the lungs vibrates the vocal fold, and at this time, a source signal with a specific frequency is generated. This source signal exits the body through the tube-shaped vocal track and spaces such as the tongue, mouth, and nose. As the shape, location, and shapes of the various organs involved in this process change over time, humans can distinguish and express various phonetic pronunciations.

To systematically understand the complex voice generation process, [11] proposes the source-filter theory. In the source-filter theory, the vibration of vocal folds generated from the lungs is defined as the source, and the influence of all organs that the source passes through until it comes out of the mouth is defined as the filter. The source-filter theory explains that the voice is generated by shaping the spectral distribution of the source through the filter.

According to the source-filter theory, the audio signal $y(t)$ is filtered through the input source signal $x(t)$ through the vocal tract, and assuming that this filter is a linear time-invariant (LTI) filter $h(t)$, the following relationship is satisfied with:

$$y(t) = h(t) * x(t) \tag{2.2}$$

where the $*$ symbol means convolution. In addition, it is expressed in the frequency domain as follows:

$$Y(\omega) = H(\omega)X(\omega) \tag{2.3}$$

where, Y, H, and X denotes the fourier transform of y, h, and x, respectively.

In this study, we applied the source-filter theory to our model by generating X and H, respectively, and then using element-wise multiplication to generate Y, which is the acoustic feature of the singing signal y. Specifically, we selected two independent decoder structures that generate source spectrum X from pitch information and filter spectrum H from pronunciation information among the input score information. We will explain in section 3 that this enabled more efficient and high-quality singing voice synthesis.

2.3 Autoregressive model

Our objective in this study is to generate a singing voice signal sequence that matches a given condition. To this end, we have to design a conditional sequence generation model. We used a conditional autoregressive model-based neural network to achieve this goal.

The autoregressive model is one of the sequential data modeling techniques. It is constructed to predict the future value based on the past value of the variable. Given time series data $x = [x_1, x_2, \ldots, x_t]$, the linear autoregressive model can be constructed with the following equation:

$$x_t = \sum_{i=t-p}^{t-1} \phi_i x_i \tag{2.4}$$

In this case, ϕ_i are the trainable variables, and p is the order of the autoregressive model.

With the development of deep learning, an attempt to parametrize the probability distribution of future values with an autoregressive model using a deep neural network in a time series prediction problem is emerging. In general, the probability distribution of $x_{1:T}$, $P(x_{1:T})$, is parameterized by the chain rule as shown in the following equation:

$$P(x_{1:T}) = \prod_{t=1}^{T} P(x_t|x_{<t}) \tag{2.5}$$

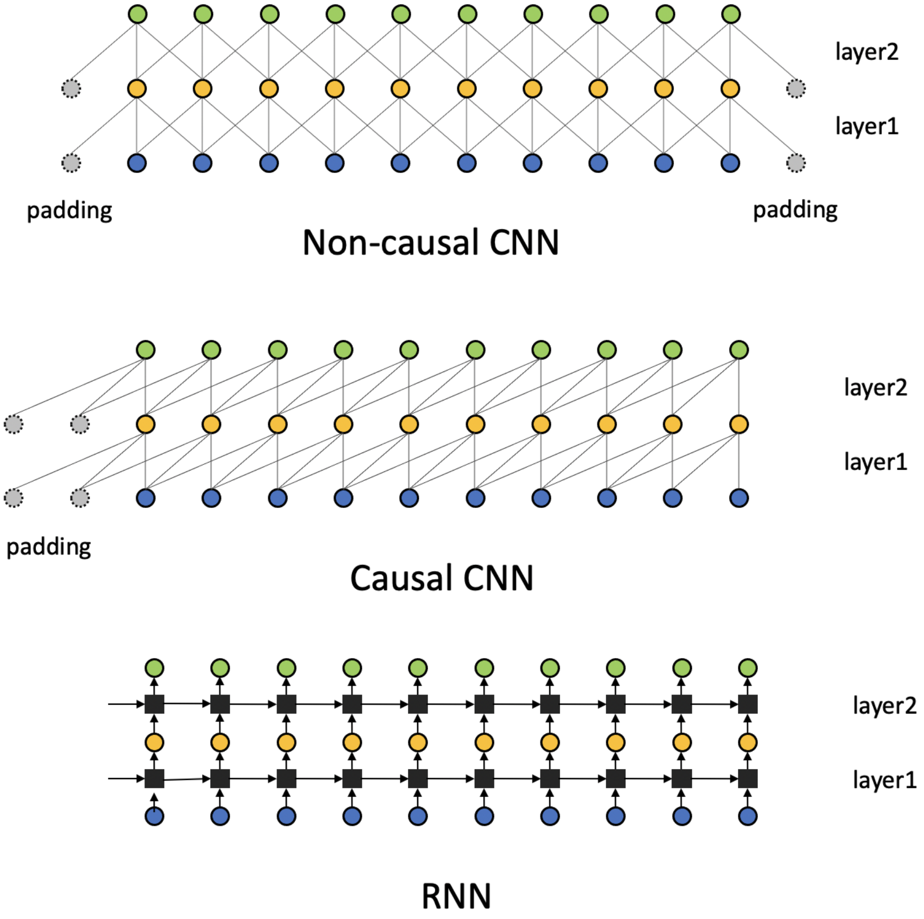
The singing voice synthesis task can be interpreted as a problem of predicting each frame of an acoustic feature according to time, and additional condition information such as the pitch, pronunciation, and singer of the score is given each frame. Condition information $c_{1:T}$ is also time-series data, and the formula of the conditional autoregressive model is as follows:

$$P(x_{1:T}|c_{1:T}) = \prod_{t=1}^{T} P(x_t|x_{<t}, c_{1:T}) \tag{2.6}$$

When designing an autoregressive model using a neural network structure, it is necessary to consider the causal characteristic of referring only to information from the past without referring to the present and future information. We can consider a recurrent neural network and a causal convolutional neural network to design a neural network that satisfies these characteristics. As shown in Figure 2.3, by controlling the padding position of the convolutional neural network, we can design the causal network and use it to parameterize the autoregressive model. In this study, we constructed a singing voice synthesis network using the dilated causal convolutional neural network structure proposed in [12].

2.4 Related works

This section reviews various speech and singing synthesis papers related to this study. In general, research on singing synthesis has a lot in common with research on speech synthesis. Both two studies are necessary to generate a human voice according to a given condition. Researchers have attempted to apply various elements found in the study of speech synthesis to singing synthesis, so looking at previous studies on speech synthesis can be a good starting point. However, there is also a difference from speech synthesis. The signal to be modeled in singing synthesis study has 1) longer syllables, 2) a wider pitch range, and 3) high-quality modeling, as shown in Figure 2.4. In addition, singing syn-



Non-causal CNN

Causal CNN

RNN

Fig. 2.3 Schematic diagram of time series data operation according to types of neural network architecture. A convolutional neural network that adds padding to both sides has non-causal properties (top). A convolutional neural network that adds padding to one side has a causal characteristic (middle). The unidirectional recurrent neural network has a casual characteristic (bottom) .

thesis has a difference in that more constraints are given as input than speech synthesis such as duration of each note, and singing style of specific singer. Therefore, we need to look at speech synthesis research by comprehensively considering the similarities and differences between the two studies. To this end, this chapter introduces the flow of various speech synthesis studies, and then introduces the singing synthesis studies conducted based on this.

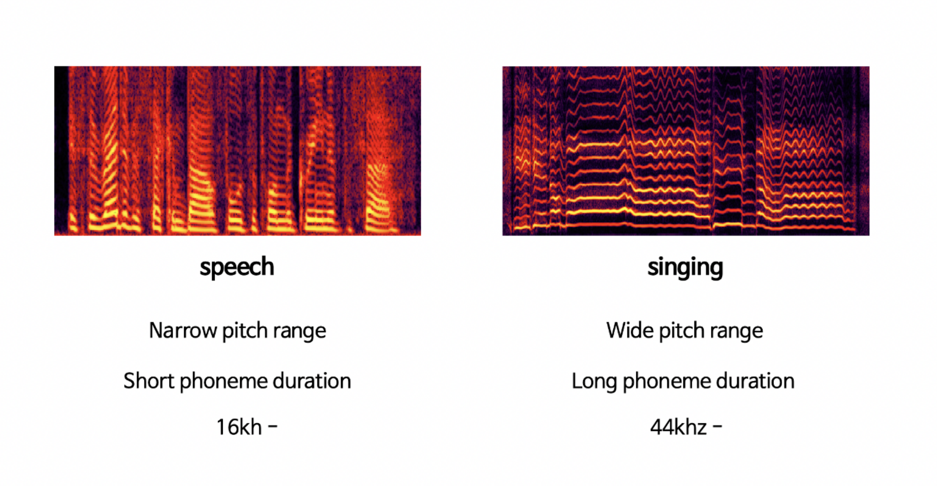


Fig. 2.4 Spectrogram image to illustrate the difference between common speech and singing signals. In general, for speech signals (left), the duration of each phoneme is short, the pitch range is not wide for the same speaker, and the frequency of requiring high-quality sound sources is low. On the other hand, in the case of singing voice (right), notes with long durations appear frequently, and even songs of the same singer sing a wider pitch range, and generally high-quality sound sources are required.

2.4.1 Speech synthesis

Speech synthesis is the task of generating a natural speech signal from a given text sequence. Prior to the advent of deep learning, speech synthesis models were generally composed of a combination of various modules as follows: a text front-end that extracts various linguistic information from input text information, a duration model that predicts the appropriate length of each phoneme, and a prosody model that predicts the appropriate pitch of a sound, and a vocoder that synthesizes speech from the generated acoustic features. In this methodology, each model was designed and trained independently, and therefore, errors generated in each model gradually accumulated. Therefore, an attempt was made to construct a speech synthesis model as a unified framework by directly modeling text-audio pairs, and end-to-end speech synthesis studies using deep learning began to proceed.

The deep learning methodology can be used as an effective tool to model the relationship between the linguistic feature obtained from the given input text and the acoustic feature obtained from the speech signal in the speech synthesis domain. A representative early end-to-end speech synthesis study is the Tacotron [13]. The Tacotron [13] is a model that autoregressively generates the linear-spectrogram of the target speech signal. At this time, the relationship between linguistic and acoustic features is modeled by using the attention mechanism with the input text signal at each generation step. The generated spectrogram is converted into a waveform through a griffin-lim vocoder, and superior MOS scores were obtained compared to the existing HMM-based parametric TTS. The Tacotron has received much attention in that it is the first model to model the entire process in an end-to-end manner by solving the modeling difficulties between sequences of different lengths of text and speech signals through an attention mechanism. However, the quality of the generated speech was not high compared to the concatenative method, and there were disadvantages such as an error in the attention mechanism for a long input text sequences. In addition, various voices cannot be generated, it is difficult to express emotion or various prosody, and the generation speed is slow due to the autoregressive nature. Therefore, starting with this, a large number of studies have been conducted to solve various problems of Tacotron.

In Tacotron 2 [14], a wavenet-based neural vocoder was introduced to improve the quality of the sound source generated by the network. Adding an autoregressive neural vocoder that generates waveforms from mel-spectrograms has been expanded to generate more realistic and close-to-real voice quality. Afterward, the researchers attempted to introduce concepts such as prosody modeling, speaker embedding, and global style token to enhance the various expressive power of the speech generation model [15, 16]. These studies, which can be viewed as an effort to secure controllability over other factors beyond text, such as timbre, prosody, style, and pitch, among various factors that determine speech, have allowed us to create more expressive speech. Furthermore, a study that can model a clean voice from noisy data by disentangling between speaker identity and recording condition [17], a study of fluent multilingual speech synthesis that can transfer the accent of a native speaker to another person's voice [18], a study to convert a dysphonia patient's voice into correct pronunciation [19], have been conducted based on the Tacotron model.

On the other hand, the autoregressive model can accumulate errors due to a structural limitation in which target acoustic features must be generated sequentially, and the generation speed is slow. Therefore, speech synthesis studies based on non-autoregressive models were also conducted to solve this limitation. A representative non-AR speech synthesis study is FastSpeech [20, 21]. In [21], a speech synthesis model that predicts each phoneme duration, pitch, and energy from the input phoneme token and performs parallel generation based on this is proposed. This has in common with the singing synthesis in that it can generate voices at high speed and secure controllability for pitch and duration. However, unlike the attention-based auto-regressive TTS system, it has limitations in that it is necessary to acquire phoneme alignment information for the speech signal in advance using an external aligner such as Montreal Forced Aligner [22].

After that, studies have emerged to eliminate the dependence on the external aligner and train the entire process from phoneme to waveform modeling end-to-end. In [23], an attempt was made to connect the latent space of mel-spectrogram and the linguistic feature by using a flow-based model. At this time, a monotonic alignment search algorithm was introduced to enable end-to-end training without an external aligner. In [24], authors proposed a more natural and high-quality end-to-end speech synthesis model by eliminating the dependence on the acoustic feature design such as a mel-spectrogram, an intermediate feature, and introducing a stochastic duration predictor.

Another important goal in speech synthesis is to design a model that generates results reflecting the styles of various speakers. This goal should also be considered in singing synthesis studies, which must understand and reflect the differences between different singers. Since the advent of Tacotron [13], which can generate a single voice, people have tried to reflect the timbre and prosody of various people. Starting with a model that can reflect the voices of various speakers by introducing speaker embedding [25], attempts are being made to effectively clone the voice using a small amount of data [26, 27, ?] or to capture the speaker's prosody [28, 29]. All of these studies can be interpreted as attempts to increase controllability and expressive power in speech synthesis system, and are methodologies that can also be applied to singing voice synthesis.

To summarize this subsection is as follows. Speech synthesis is one of the representative tasks that share a goal similar to singing voice synthesis and has been developed in various directions along with deep learning. The researchers first designed an end-to-end framework that generated audio from the input text and tried to improve the synthesis quality of this model. In addition, to synthesize a more expressive voice, researchers conducted various studies to design a system that can capture the characteristics of different voices and control them. Many parts of the research on singing synthesis have been developed based on the concepts proposed in the research on speech synthesis. In particular, in this study, a method of parametrizing an autoregressive model using a neural network was applied, which helped design a model that can keep the characteristics of a singing signal well consistent. In addition, by extending the concept of global style token introduced to capture the speaker's characteristics, we proposed a local style token module that can model the expression elements of singing signals according to the passage of time. As such, the study on speech synthesis was of great help in that it introduced and suggested the basic concepts of many studies of singing synthesis.

2.4.2 Singing voice synthesis

Singing voice synthesis research proceeded with the development of speech synthesis research. The representative early SVS study to which a neural network-based methodology is applied is [30]. In [30], the authors proposed an SVS system as a method of generating parametric vocoder features from input score using a neural network based on a wavenet[12] architecture. This study is significant because it established standards for the music score representation method for singing synthesis task and training and evaluation of a neural network-based model. However, the model proposed in this study has a limitation in that it is limited by the performance limit of the parametric vocoder.

Most of the follow-up studies related to singing synthesis have been developed to synthesize better-quality singing by exploring the structure of the neural network. Researchers have tried to obtain better sound quality of synthesized singing voice by using various neural network structures such as CNN[31, 32], RNN[33, 34], GAN [35, 36], Transformer[37], VAE [38], and Diffusion probabilistic model [39] for the singing synthesis task. In addition, in order to overcome the limitation of insufficient data for model training, studies have been actively conducted, such as, to use data mined from the web for training[40], train with speech data[41, 18, 42], or consider a more efficient voice cloning method[43]. In order to obtain an advantage in terms of generation speed, studies using non-autoregressive structures[44] or MLP structures[45] have also been conducted. Apart from performance improvement, new tasks related to singing synthesis have been proposed and studied. For example, unconditional singing generation in the absence of sheet music[46] or research to generate rapping voice[47] is an example.

In summary, singing synthesis studies have been developed to explore and apply various neural network model architectures along with the development of speech synthesis research. The main direction is to improve the sound quality of synthesized singing voices. In addition, studies were conducted considering the speed and data efficiency of the system itself. However, to the best of our knowledge, there are not yet many studies that have considered models with high usability while producing more artistically expressive results. This study focuses on designing a system that helps create more expressive and artistic results with high usability based on the various techniques discovered by various speech synthesis and singing synthesis research.

Chapter 3

Adversarially Trained End-to-end Korean Singing Voice Synthesis System

3.1 Introduction

With the recent development of deep learning, a learning-based singing voice synthesis (SVS) system, which synthesizes sounds as natural as the concatenative method [48, 49, 50], but can expand more flexibly, is proposed. For example, three SVS systems based on DNN, LSTM, and Wavenet architecture were proposed, respectively [51, 33, 30]. These systems all include an acoustic model that is trained by singing, lyrics, and sheet music paired data, and each acoustical model is trained to predict the vocoder feature used as an input to the vocoder.

Although these neural network-based SVS system can achieve adequate performance, networks predicting vocoder features have limits that cannot exceed the upper bound of vocoder performance. Therefore, it is meaningful to propose an end-to-end framework that directly generates a linear-spectrogram, not a vocoder feature. However, the extension to the end-to-end framework of the SVS system is a challenging task because it involves increased complexity of the model. Creating a more complex target, linear-spectrogram, increases the complexity of the model and requires as much training data to generalize and train these models sufficiently. However, gathering singing audio with aligned lyrics in a controlled environment is a task that requires a lot of effort.

We proposed in this paper a Korean SVS system that can be trained by an end-to-end manner with moderate amounts of data [1]. Our baseline network is designed with the inspiration of DCTTS [1], known as efficiently trainable text to speech (TTS) system. We applied the following novel approaches to enable end-to-end network training. First, we used the phonetic enhancement masking method, which separately modeled low-level acoustic features related to pronunciation from text information, to make more efficient use of the information contained in the training data. Second, we also proposed a method of reusing input data at the super-resolution stage and training with an adversarial manner to produce better sound quality singing.

The contribution of this paper is as follows: **1)** We designed the end-to-end Korean SVS system and suggested a way to train it effectively. **2)** We proposed a phonetic enhancement masking method that helps to produce more accurate pronunciation. **3)** We proposed a conditional adversarial training method for the generation of more realistic singing voices.

[1] The generated result can be found at: ksinging.strikingly.com.

3.2 Related work

The SVS system is similar to the TTS system in terms of synthesizing natural human speech. Recently, the end-to-end TTS system, which is trained as an autoregressive manner, such as Tacotron[13], Deep voice[25], is showing better performance than the conventional method. In addition, various follow-up studies are being conducted that have further controllable elements such as prosody, style, etc [15, 16], or models that can be trained more efficiently [1, 52]. We conducted the study by modifyng the TTS model to suit the SVS task, based on DCTTS[1], which is known to be capable of efficient end-to-end training.

The generative adversarial networks (GAN) is a widely used technique that helps train an arbitrary function to generate a similar sample as the sample from desired data distribution. This training method has been widely accepted in computer vision community and becomes one of the key components to attain photo-realism in super-resolution task. Unlike the success of adversarial training method in image domain, however, only a few works have achieved a reasonable success of training super-resolution task (specifically, band-width extension task) in audio signal processing community [53]. To further leverage the promise of adversarial training in audio generation process, we adopted a few recent works that stabilizes the adversarial training, namely, conditional GAN with projection discriminator [54] and R1 regularization [55] which allow us to jointly train the autoregressive network (mel-synthesis) and super-resolution network making the proposed system as an end-to-end framework.

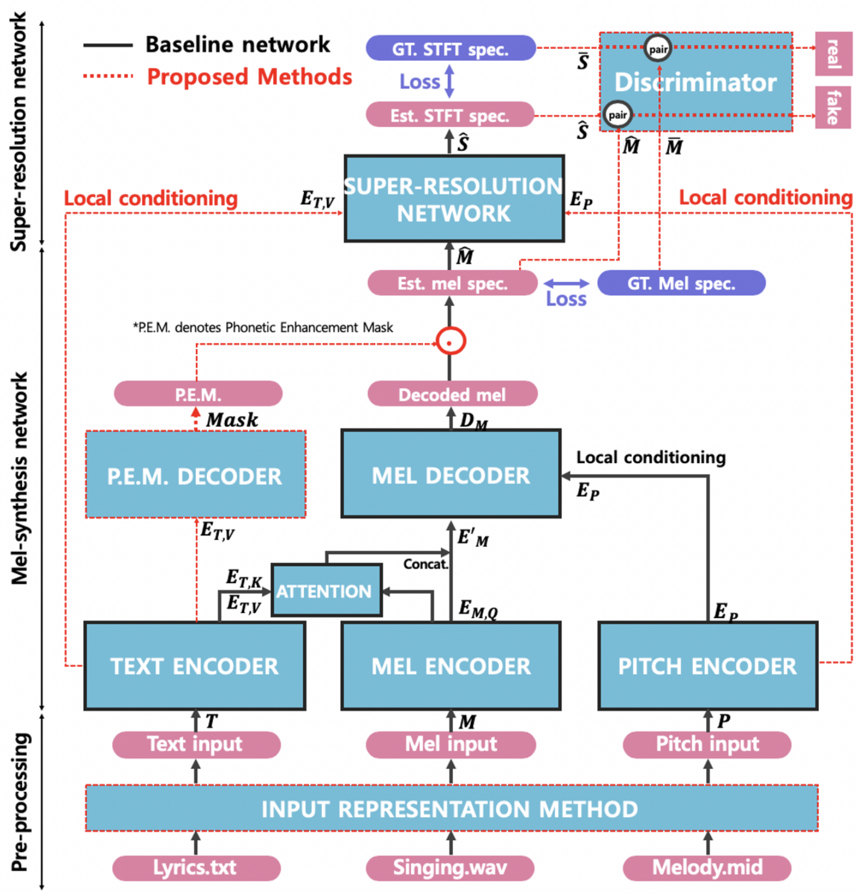


Fig. 3.1 An overview of the proposed single-singer singing voice synthesis system designed based on the source-filter theory.

3.3 Proposed method

As illustrated in Figure 3.1, our proposed model consists of two main modules, a mel-synthesis network and a super-resolution network. The mel-synthesis network is trained to produce a mel-spectrogram $M_{1:L}$ from previous mel input $M_{0:L-1}$, time-aligned text $T_{1:L}$, and pitch inputs $P_{1:L}$. With text and pitch information as conditional input, the super-resolution network upsamples the generated mel-spectrogram M to a linear-spectrogram S. Finally, the discriminator takes the upsampled result with generated mel-spectrogram to train the network in an adversarial manner.

During the test phase, a sequence of mel-spectrogram frames is generated in an autoregressive manner from a given text and pitch input which is then upsampled to linear-spectrogram by super resolution network. Finally, the generated linear-spectrogram is converted to a waveform using Griffin-Lim algorithm [56].

3.3.1 Input representation

Our training data includes recorded singing voice along with the corresponding text and midi. A single midi note represents pitch information with onset and offset. For the single midi note, one syllable and its corresponding vocal audio section are manually aligned. Figure 3.2 shows our input representation more concretely.

To determine the text input sequence $T \in R^{1 \times L}$ with length L, we referred to the pronunciation system of Korean. A Korean syllable can be decomposed into three phonemes each of which corresponds to onset, nucleus, and coda, respectively.

Since the nucleus occupies most of the pronunciation singing in Korean, we assigned onset and coda to the first and the last frame of input text array, respectively, and the rest of the frames with nucleus.

Although this does not reflect accurate timing for each phoneme, we empirically found out that a convolution-based network with wide enough range of receptive field can handle this problem. For pitch input $P \in R^{1 \times L}$, we simply assigned a pitch number to each frame. In the case of the mel input $M \in R^{F \times L}$, we used the mel-spectrogram itself, which was extracted from the recorded audio, where F denotes the number of frequency bins.

3.3.2 Mel-synthesis network

The mel-synthesis network $MS(\cdot)$ aims to generate the mel-spectrogram of the next time step from the given text, pitch, and mel input. Based on the text-to-mel network proposed by [1] we modified it to fit the SVS system.

First, in order to enter pitch information, we added pitch encoders with the same structure as text encoders. In addition, the local conditioning method proposed by [12] was used to conduct a conditioning of the encoded pitch on the mel decoder.

Second, we assumed that among the various elements forming a singing voice, information about pronunciation would be able to be controlled independently from text information. We also assumed that if the low-level audio feature that constitutes pronunciation information can be modeled independently, it is possible to focus on the pronunciation information in the data composed of various combinations of pronunciation-pitch, so that training data can be utilized more efficiently to generate more accurate pronounced singing

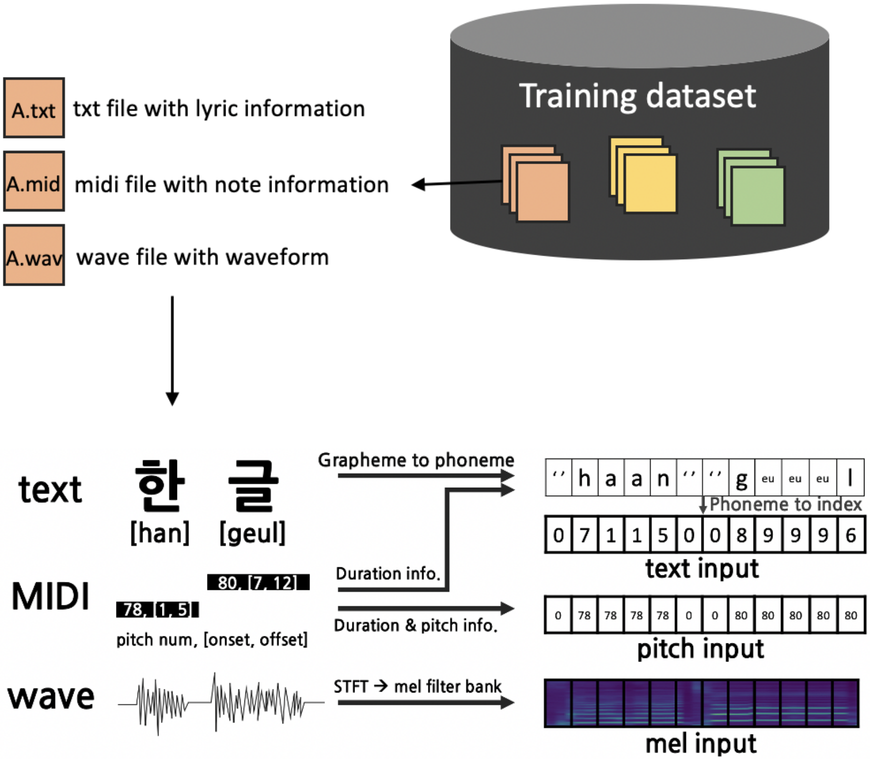


Fig. 3.2 A schematic diagram representing the components of the collected training singing dataset and the transformation process into an input representation.

voice. To this end, we designed an additional phonetic enhancement mask decoder, which receives encoded text only as input, and the output of the decoder element-wise multiplied by the output of the mel decoder to create the final mel-spectrogram. As a result, $MS(\cdot)$ can be formulated as follows:

$$\hat{M} = Mask \odot D_M = MS(M, T, P) \tag{3.1}$$

We trained the $MS(\cdot)$ network with L_1 and binary divergence loss L_d between ground truth and generated mel-spectrogram, and guided attention loss L_{att} as the objective function. Please see [1] for more detailed explanation on the loss terms.

We also assumed that the L_1 loss between the differential spectrogram $M' = M_{1:L} - M_{0:L-1}$ would be also beneficial for network to learn more about the relatively short pronounced onset, coda. Therefore, the overall objective function for $MS(\cdot)$ is as follows:

$$L_{MS} = L_1(\hat{M}, M) + L_d(\hat{M}, M) + L_{att} + L_1(\hat{M}', M') \tag{3.2}$$

3.3.3 Super-resolution network

In this section, we describe the details of the training method for super-resolution network $SR(\cdot)$. The purpose of the SR step is to upsample the generated mel-spectrogram $\hat{M} \in R^{F \times L}$ into a linear-spectrogram $\hat{S} \in R^{F' \times L'}$ thereby making it to an audible form, where F' and L' denote the number of frequency bins and temporal bins for linear-spectrogram. The idea of the SR network was proposed in a few previous TTS literatures, including Tacotron and its variants [1, 13]. The major difference between the previous works and our work is twofold. First, we additionally reuse the aligned text and pitch information into the SR network exploiting the useful information in the generation process again. Second, we utilize adversarial training methods to make the SR network produce more realistic sound.

Local conditioning of text and pitch information

Unlike the attention-mechanism based TTS literature, SVS system requires the aligned text and pitch information as inputs for the controllability in the generation process. These information, therefore, can be easily reused in the SR step in the absence of time-alignment process as follows.

$$\hat{S} = SR(\hat{M}, E_{T,V}, E_P) = SR(MS(\cdot), E_{T,V}, E_P) \tag{3.3}$$

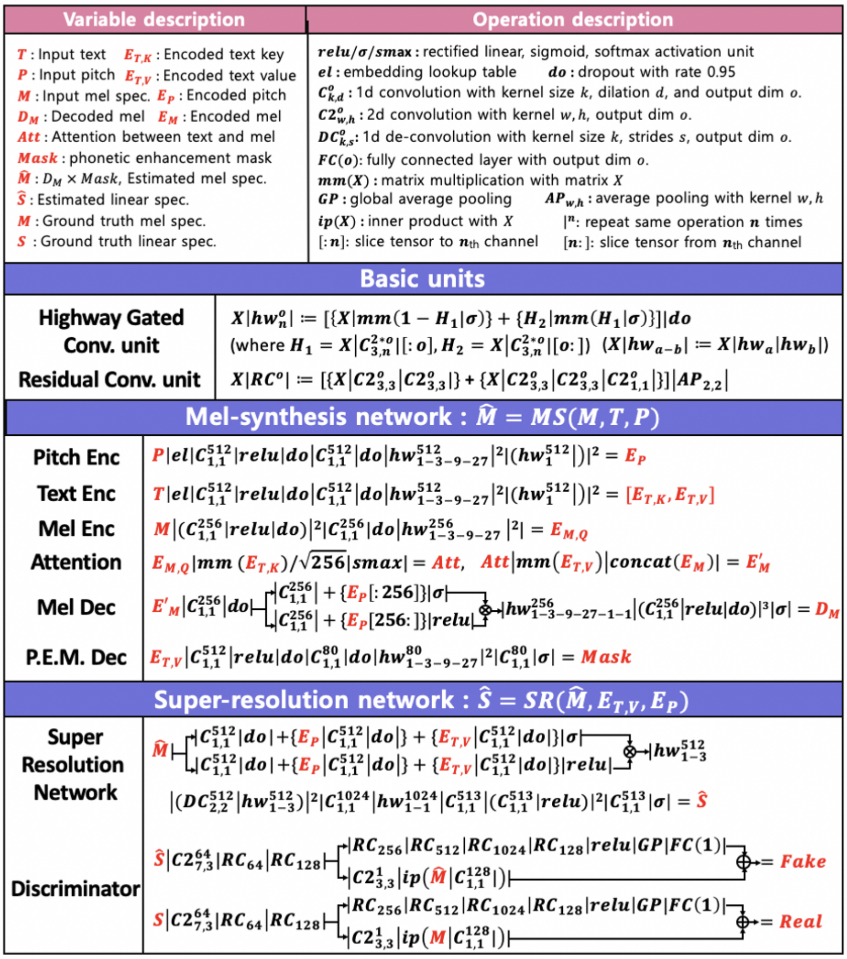
More specifically, each of the output from the text encoder and pitch encoder ($E_{T,V}$ and E_P) is fed into a sequence of 1×1 convolutional and dropout layer [57] which is then fed into a highway network as a local conditioning method as proposed in [12]. For the upsampled $\hat{S}$, SR is trained with the objective function $L_{SR} = L_1(\hat{S}, S) + L_d(\hat{S}, S)$. For the exact network configuration, please refer to Figure 3.3.

Adversarial training method

Expecting to generate a realistic sound, we adopted a conditional adversarial training method which helps the output distribution of $\hat{S} = SR(\hat{M}, \cdot)$ be similar to the real data distribution $S \sim p(S|M)$.

Intuitively, in the conditional adversarial training framework, discriminator D_ψ not only tries to check if S is realistic but also the paired correspondence between S and M.

Note that, we make a minor assumption that the distribution of $\hat{M} = MS(\cdot)$



Variable description

T : Input text $E_{T,K}$: Encoded text key

P : Input pitch $E_{T,V}$: Encoded text value

M : Input mel spec. E_P : Encoded pitch

D_M : Decoded mel E_M : Encoded mel

Att : Attention between text and mel

$Mask$: phonetic enhancement mask

$\widehat{M}$: $D_M \times Mask$, Estimated mel spec.

$\widehat{S}$: Estimated linear spec.

M : Ground truth mel spec.

S : Ground truth linear spec.

Operation description

$relu/\sigma/smax$: rectified linear, sigmoid, softmax activation unit

el : embedding lookup table do : dropout with rate 0.95

$C_{k,d}^{o}$: 1d convolution with kernel size k, dilation d, and output dim o.

$C2_{w,h}^{o}$: 2d convolution with kernel w, h, output dim o.

$DC_{k,s}^{o}$: 1d de-convolution with kernel size k, strides s, output dim o.

$FC(o)$: fully connected layer with output dim o.

$mm(X)$: matrix multiplication with matrix X

GP : global average pooling $AP_{w,h}$: average pooling with kernel w, h

$ip(X)$: inner product with X $|^n$: repeat same operation n times

$[:n]$: slice tensor to n_{th} channel $[n:]$: slice tensor from n_{th} channel

Basic units

Highway Gated Conv. unit

$$X|hw_n^o| \coloneqq [\{X|mm(1 - H_1|\sigma)\} + \{H_2|mm(H_1|\sigma)\}]|do$$

(where $H_1 = X\left|C_{3,n}^{2*o}\right|[:o], H_2 = X\left|C_{3,n}^{2*o}\right|[o:]$) $(X|hw_{a-b}| \coloneqq X|hw_a|hw_b|)$

Residual Conv. unit

$$X|RC^o| \coloneqq [\{X|C2_{3,3}^o|C2_{3,3}^o|\} + \{X|C2_{3,3}^o|C2_{3,3}^o|C2_{1,1}^o|\}]|AP_{2,2}|$$

Mel-synthesis network : $\widehat{M} = MS(M, T, P)$

Pitch Enc

$$P|el|C_{1,1}^{512}|relu|do|C_{1,1}^{512}|do|hw_{1-3-9-27}^{512}|^2|(hw_1^{512}|)|^2 = E_P$$

Text Enc

$$T|el|C_{1,1}^{512}|relu|do|C_{1,1}^{512}|do|hw_{1-3-9-27}^{512}|^2|(hw_1^{512}|)|^2 = [E_{T,K}, E_{T,V}]$$

Mel Enc

$$M|(C_{1,1}^{256}|relu|do)|^2|C_{1,1}^{256}|do|hw_{1-3-9-27}^{256}|^2| = E_{M,Q}$$

Attention

$$E_{M,Q}|mm\,(E_{T,K})/\sqrt{256}|smax| = Att, \quad Att|mm(E_{T,V})|concat(E_M)| = E'_M$$

Mel Dec

$$E'_M|C_{1,1}^{256}|do| \begin{cases} |C_{1,1}^{256}| + \{E_P[:256]\}|\sigma| \\ |C_{1,1}^{256}| + \{E_P[256:]\}|relu| \end{cases} \otimes |hw_{1-3-9-27-1-1}^{256}|(C_{1,1}^{256}|relu|do)|^3|\sigma| = D_M$$

P.E.M. Dec

$$E_{T,V}|C_{1,1}^{512}|relu|do|C_{1,1}^{80}|do|hw_{1-3-9-27}^{80}|^2|C_{1,1}^{80}|\sigma| = Mask$$

Super-resolution network : $\widehat{S} = SR(\widehat{M}, E_{T,V}, E_P)$

Super Resolution Network

$$\widehat{M}| \begin{cases} |C_{1,1}^{512}|do| + \{E_P|C_{1,1}^{512}|do|\} + \{E_{T,V}|C_{1,1}^{512}|do|\}|\sigma| \\ |C_{1,1}^{512}|do| + \{E_P|C_{1,1}^{512}|do|\} + \{E_{T,V}|C_{1,1}^{512}|do|\}|relu| \end{cases} \otimes |hw_{1-3}^{512}$$

$$|(DC_{2,2}^{512}|hw_{1-3}^{512})|^2|C_{1,1}^{1024}|hw_{1-1}^{1024}|C_{1,1}^{513}|(C_{1,1}^{513}|relu)|^2|C_{1,1}^{513}|\sigma| = \widehat{S}$$

Discriminator

$$\widehat{S}|C2_{7,3}^{64}|RC_{64}|RC_{128}| \begin{cases} |RC_{256}|RC_{512}|RC_{1024}|RC_{128}|relu|GP|FC(1)| \\ |C2_{3,3}^{1}|ip(\widehat{M}|C_{1,1}^{128}|)| \end{cases} \oplus = Fake$$

$$S|C2_{7,3}^{64}|RC_{64}|RC_{128}| \begin{cases} |RC_{256}|RC_{512}|RC_{1024}|RC_{128}|relu|GP|FC(1)| \\ |C2_{3,3}^{1}|ip(M|C_{1,1}^{128}|)| \end{cases} \oplus = Real$$

Fig. 3.3 Detailed structure of each sub-module. $X|F|$ denotes $F(X)$

approximately follows that of M, that is, $p(M) \simeq p(\hat{M})$, allowing the joint training of two modules $MS(\cdot)$ and $SR(\cdot)$. The conditioning to discriminator was done by following [54] with a minor modification. First, the condition M is fed into a 1d-convolutional layer and the intermediate output of discriminator is fed into a 3×3 2d-convolutional layer. Then, inner product between the two outputs is done as a projection. Finally, the obtained scalar value is added to the last layer of D_ψ resulting in final logit value. For the exact network configuration please refer to Figure 3.3.

For the stable adversarial training, a regularization technique on D_ψ has been proposed by several GAN related works [58, 59, 60, 55]. We adopted a simple, yet, effective gradient penalty technique called R1 regularization. This technique penalizes the squared 2-norm of the gradients of D_ψ only when the sample from true distribution is taken as follows

$$R_1(\psi) = \frac{\gamma}{2} E_{p(M,S)}[\nabla D_\psi(M,S)^2]. \tag{3.4}$$

Note that the output of D_ψ denotes the logit value before the sigmoid function. The final adversarial loss terms (L_{adv_D} and L_{adv_G}) for D_ψ and G_θ are as follows,

$$\begin{aligned} L_{adv_D}(\theta,\psi) &= -E_{p(M)}[E_{p(S|M)}[f(D_\psi(M,S))]] \\ &\quad - E_{p(\hat{M})}[f(D_\psi((\hat{M},\hat{S}))] + R_1, \\ L_{adv_G}(\theta,\psi) &= E_{p(\hat{M})}[f(D_\psi(\hat{M},\hat{S}))], \end{aligned} \tag{3.5}$$

where θ includes not only the parameters of SR but also that of MS, hence the two consecutive modules acting as one generator function $G_\theta = SR(MS(\cdot),\cdot)$. The function f is chosen as follows $f(t) = -log(1 + exp(-t))$ resulting in the vanilla GAN loss as in the original GAN paper [61].

3.4 Experiments

3.4.1 Dataset

Since there is no publicly available Korean singing voice dataset, we created the dataset as follows. First, we prepared accompaniment and singing voice MIDI files of 60 Korean pop songs. Next, a professional female vocalist was told to sing to the accompaniment. Then, the singing voice MIDI files were manually realigned so that the recorded audio have the exact alignment with the singing voice MIDI files. Finally, we manually assigned the syllables in lyrics to each MIDI note of singing voice MIDI file. The audio length of the entire dataset excluding the silence is about 2 hours. We used 49 songs for training dataset, 1 song for validation, and 10 songs for test dataset.

3.4.2 Training

We trained the discriminator to minimize L_{adv_D} and the rest of the network to minimize L_{adv_G}, L_{MS} and L_{SR}. For SR networks, we have to start training after the appropriate level of mel is generated, so we have separately controlled lr_{SR} and lr_{GAN} to add to the objective function. At this point, it was set to $lr_{SR} = \min(0.2 * (\text{iter}/100), 1)$, $lr_{GAN} = \min(0.01 * (\text{int})(\text{iter}/5000), 1)$, respectively.

$$\begin{aligned} L_{MS,SR} &= L_{MS} + lr_{SR} \cdot L_{SR} + lr_{GAN} \cdot L_{adv_G} \\ L_D &= lr_{GAN} \cdot L_{adv_D} \end{aligned} \tag{3.6}$$

In both cases, we used Adam optimizer [62], which was set to $\beta_1 = 0.5$ and $\beta_2 = 0.9$. The learning rate was scheduled to start from 0.0002 and was halved for every 30,000 iteration. All parameters of the networks were initialized with the Xavier initializer [63].

For the ground truth mel/linear-spectrogram, we first extracted the linear-spectrogram S from audio with $sr = 22050, n_{fft} = 1024, hop = 256$. We then normalized the linear-spectrogram as follows $S \leftarrow (|S|/max(|S|))^{\delta}$, where δ denotes a pre-emphasis factor with the value of 0.6 in our case. [2]

Afterwards, the mel-spectrogram was obtained by multiplying 80-d of mel filter bank to S, and the same normalization method as in S was used. In order to reduce the complexity of the model, we downsampled the mel-spectrogram to the quarter by taking the first frame of every four-frame of the mel-spectrogram giving the relationship of $L' = 4L$.

3.4.3 Evaluation

We trained a total of five models to see how the three proposed methods - method 1; phonetic enhancement masking, method 2; local conditioning pitch and text to $SR(\cdot)$, method 3; adversarial training method - actually affect the network. The differences between the five models are described in Table 3.1. 20 audio samples from each model were generated from the test dataset. Apart from the generated samples, we also compared the ground truth samples. **Ground** denotes the actual recorded audio, and **Recons** denotes the reconstructed audio from ground truth magnitude only linear-spectrogram using Griffin-Lim algorithm. Noe that **Recons** samples were included to evaluate the sound quality from the loss of phase information.

[2] Note that we post emphasized $\hat{S} \leftarrow \hat{S}^{\zeta/\delta}$ where ζ denotes a post-emphasis factor with the value of 1.3.

Table 3.1 A description of the models that have been tested for comparison to verify the performance of the proposed techniques

Model	model1	model2	model3	model4	model5
Method	baseline	+(method 1)	+(method 2)	+(method 1,2)	+(method 1,2,3)

Quantitative evaluation

We evaluated whether the network was actually producing a conditioned singing voice for a given input. To do this, we extracted f0 sequence from the generated audio through the world vocoder[64], converted it into a pitch sequence, and compared it to the input pitch sequence. We can judge that the higher the similarity between the two sequence, the more the network generates a singing that reflects the input condition. We calculated the precision, recall and f-score of the generated pitch sequence by frame-wise, and the results are shown in Table 3.2.

Even in the case of a real recording sample recorded by listening to the original midi accompaniment, it is not easy to adjust the timing and pitch of the correct note, so that a 100% accurate f-score can not be obtained. For all samples that were generated, a f-score similar to or higher than the real recording sample was obtained. This means that the model has generated a singing voice with the correct pitch and timing for at least the real recording for the given input.

Qualitative evaluation

We conducted a listening test to evaluate the quality of the generated singing voice. 19 native Korean speakers were asked to listen to the 20 audio samples from each model.

Table 3.2 Experimental results obtained from the proposed models. For quantitative evaluation, the f0 coincidence rate was measured to see if a result that reflected the input score well was generated. For qualitative evaluation, listening evaluation was conducted on pronunciation accuracy, sound quality and naturalness.

	Quantitative			**Qualititative**		
Model	Precision	Recall	F1-score	Pronun.acc	Sound.quality	Naturalness
model1	0.771	0.832	0.800	2.29 ±1.15	2.32 ±0.89	2.11 ±0.98
model2	0.780	**0.843**	0.810	2.62 ±1.00	2.28 ±0.84	2.22 ±0.91
model3	0.755	0.814	0.783	2.69 ±1.06	2.37 ±0.86	2.22 ±0.93
model4	0.792	0.832	0.811	2.92 ±1.08	2.43 ±0.86	2.36 ±0.94
model5	**0.872**	0.821	**0.846**	**3.23 ±1.19**	**3.37 ±0.94**	**3.07 ±1.10**
Recons	0.805	0.830	0.782	4.85 ±0.47	4.46 ±0.77	4.72 ±0.62
Ground	0.826	0.772	0.798	4.90 ±0.36	4.74 ±0.57	4.85 ±0.43

Each participant was asked to evaluate the pronunciation accuracy, sound quality, and naturalness. During the listening test, lyrics of audio samples were provided for more accurate evaluation of pronunciation accuracy. The MOS results are shown in Table 3.2.

We conducted a paired t-test for each model response and based on this we verified the effectiveness of the proposed methods. For the accuracy of the pronunciation, we obtained significant differences for all comparisons except for models 2 and 3. In other words, all of the proposed methods helped to create more accurate pronunciation singing voices, and the performance was improved to the greatest extent with all three methods. In the case of sound quality, methods 1 and 2 did not significantly affect the improvement, but the applying method 3 showed a significant increase in score. From this we can confirm that training the network in an adversarial manner improves the quality of the generated audio. Finally, for naturalness, there was a significant improvement when all methods were applied.

3.4.4 Analysis on generated spectrogram

In this section we analyze the features generated by the mel-synthesis and super-resolution networks. In the case of mel-synthesis network, from observing internally generated features, we found that the low-level acoustic feature of pronunciation and pitch could be divided independently without any supervision. From Figure 3.4, D_M shows the underlying structure of the spectrogram, such as the harmonic structure and the location of f0. In $Mask$, on the other hand, we can observe the shape of determining the intensity of the frequency at every time-step, similar to the feature of the spectral envelope, which contains non-periodic information. This suggests that, from the perspective of source-filter models, one of the techniques that classical speech modelling techniques, our network can generate sources (D_M) and filters ($Mask$) separately from frequency domain without any supervised training.

We also analyzed the effect of adversarial training method by observing the generated linear-spectrogram. Three different spectrograms from model4 ($\hat{S}'$: w/o adversarial loss), model5 ($\hat{S}$: w/ adversarial loss), and ground truth spectrogram ($\bar{S}$) are demonstrated in the second row of Figure 3.5. While $\hat{S}'$ showing the blurry high frequency areas, $\hat{S}$ clearly shows that adversarial training allows the proposed network to generate sample that is closer to the ground truth sample $\bar{S}$. Note that we have confirmed in 3.4.3, listening test that the sound quality can be significantly improved by comparing model4 and model5, which again reinforces our observation.

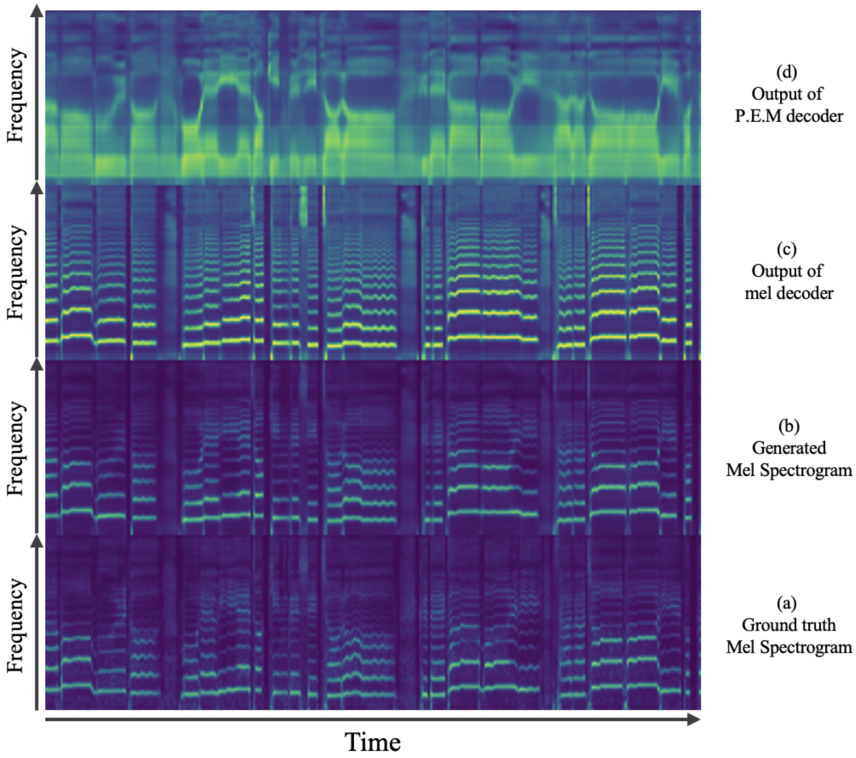


Fig. 3.4 Spectrogram of the generated singing signal and the ground truth singing signal. It can be seen that two different decoders generate information on pronunciation and pitch, respectively. It can be seen that the product of the two outputs is similar to the actual ground truth spectrogram.

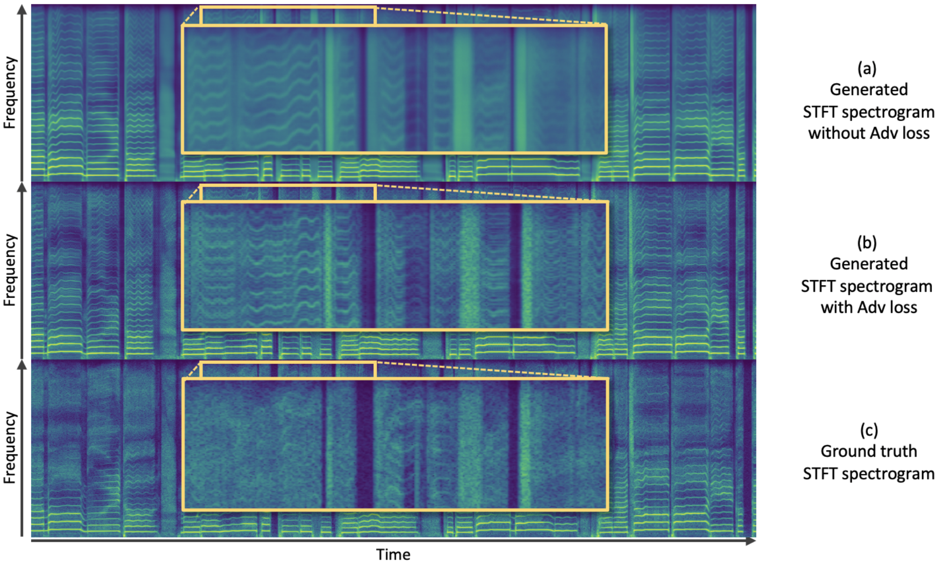


Fig. 3.5 Comparison result of spectrogram generated according to whether adversarial training is applied. When adversarial training is applied, it can be seen that the detailed expression of the ground truth spectrogram is similarly generated compared to the case where adversarial training is not applied.

3.5 Discussion

3.5.1 Limitations of input representation

In this study, we performed note-level annotation on a given singing voice dataset to obtain the training data. In note-level annotation, one note corresponds to one syllable in the audio. The annotator directly listened to the singing voice, performed note unit transcription, and then assign syllables and pitches to each note. For more detailed modeling, more detailed phoneme-level annotation than syllable is required. However, phoneme-level annotation was not carried out in this study for two reasons. First, it consumes much time to classify and transcribe data in phoneme units. Unlike syllables, phonemes are uttered in a very short moment, and it is difficult for people unfamiliar with the work to distinguish them properly. Second, we judged that the use of phoneme-level input representation would degrade SVS usability for users. In making music using SVS, it would be time-consuming if the producer had to input every phoneme sequence one by one. Instead, inputting syllable-by-syllable sheet music will allow for greater convenience for producers to properly generate results that consider the arrangement of phoneme units by themselves. Therefore, we designed a singing synthesis system that uses syllable unit input instead of phoneme unit input for the above two reasons. Then, as mentioned in section 3.3.1, an input representation method was used that assigns one frame each to onset and coda phoneme and nucleus to the remaining frames among the ranges occupied by each note.

However, there are some problems with this input representation method. First, since we do not use phoneme-level alignment, a timing mismatch between the input information and the target signal occurs, as shown in Figure 3.6.

Based on the fact that vowels occupy the longest time in a syllable, we assign most note durations to vowel durations. However this is not always a correct assumption. In the case of a note with a long duration, there are cases where the length of the coda occupies a lot of weight. Also, certain voiced consonants are pronounced longer than one frame. For this reason, a mismatch occurs because our proposed model is trained to generate speech from not perfectly matched score input. Second, our note-level annotations are not perfect either. An annotator can annotate a note at a position faster or slower than the onset of the actual sound source. The pitch may be different, and the pronunciation may be misspelled. Therefore, in our training data, 1) there is no alignment for phoneme units, and 2) there may be differences in duration, phoneme, and pitch from the actual sound source.

Nevertheless, we confirmed that the proposed model could synthesize a singing voice that faithfully reflects the input score resulting from the experiment. In particular, as a result of analyzing the degree of agreement between the pitch of the input MIDI and the pitch of the generated output, as described in section 3.4.3, the agreement rate was higher than that of the actual data. We considered two possible reasons for this result. The first reason is that our model consists of an auto-regressive structure. Instead of a non-AR structure that generates singing voices directly from input information extracted from sheet music, our model sequentially generates spectrograms in an autoregressive manner, referring to sheet music information as conditional input. Therefore, as the degree of dependence of the model on sheet music information in the generation process is relatively low, it is interpreted that modeling was possible even when conditional input was not given according to precise timing. Second,

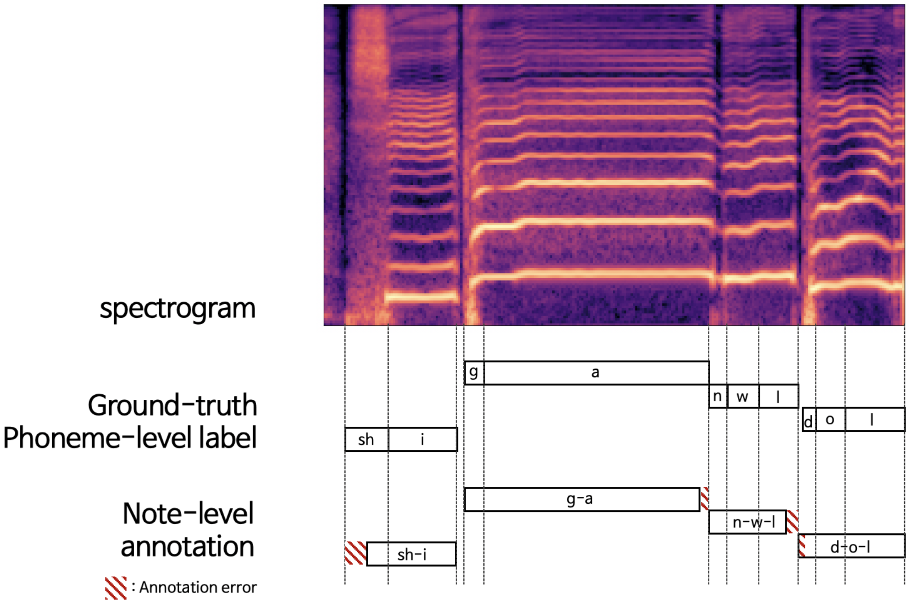


Fig. 3.6 Illustration of singing voice spectrogram, ground truth phoneme-level label, and note level annotation by annotator. Mismatch occurred between the actual singing voice and the onset and offset timing in note-level annotation due to an error that occurred by the annotator. Also, since only the boundary for the syllable unit is annotated, more detailed phoneme-level alignment information cannot be obtained.

smoothing for error may be the reason. Assuming that the error generated by the annotator over various data follows a Gaussian distribution, the average value of the data is likely to be the correct label. Therefore, in the generative modeling methodology based on repetitive training, it can be interpreted that the training proceeded in the direction of offsetting these errors.

Nevertheless, the following two considerations are necessary for better singing synthesis modeling. First, from the model's point of view, input information should be provided in a more accurate form. It is desirable to obtain detailed alignment in phoneme sequence and use pitch and text labels with less error. For this, alignment between data and score can be obtained using forced alignment techniques. However, as a result of internal experiments, it was confirmed that the alignment tools designed for speech generally do not work well for singing. This seems to be because singing includes a longer duration, pronunciation changes, and a variety of pitch compared to common speech voices. Therefore, additional research on an improved alignment tool to work well in singing voice is needed.

The second is about usability. It is inconvenient for users to create and input sheet music at the phoneme level. For higher usability, when a note-level expression composed of syllable units is input, the model's front-end appropriately converts it into a phoneme-level representation. However, unlike the duration model used in general non-AR TTS, this is a problem in which each phoneme's detailed length and position must be determined in the presence of a constraint for the entire syllable length. Therefore, it is necessary to study a duration model for a singing synthesis model that can predict the length of the phoneme sequences inside a note in the presence of constraints.

3.5.2 Advantages of using super-resolution network

We designed our singing synthesis system based on the [1]. Our proposed model generates a linear spectrogram and finally generates a 22050 Hz waveform through a griffin-lim vocoder. The hop size of the linear spectrogram modeled by the acoustic model is 256, so the model is trained to generate frame-level acoustic features of about 86Hz ($\simeq 22050/256$) time-resolution. Our model includes a super-resolution network that upsamples the time-resolution by a factor of 4, so the acoustic model before the super-resolution network generates a frame-level feature of 21.5 Hz ($= 86/4$). The network structure that generates a mel-spectrogram of lower time resolution first and then upsamples them has three advantages.

First, it has an advantage in terms of inference speed. Since our model is auto-regressive, mel-spectrograms must be continuously generated through sequential iteration. Therefore, the time required for generation is proportional to the length of the target to be created, and most of the time is spent in auto-regressive iteration. Our model first generates a spectrogram with four times lower time resolution and then upsamples the generated result in parallel so that it can be generated about four times faster.

Second, there is an advantage that the influence of errors caused by human annotators can be reduced. As mentioned in the previous section, our training data has subtle timing errors. These errors can act as a hindrance to training. However, we can reduce the relative proportion of these errors by modeling the mel-spectrogram with a lower time resolution. For example, suppose there is an error of 40 ms between the onset of the singing audio and the MIDI label in the 86 Hz representation. In that case, one frame has a length of about

11 ms, so there is a difference of about four frames. However, in the 21.5 Hz representation, one frame has a length of about 44ms. Only a difference of about one frame occurs. Therefore, more robust training against timing errors is possible when modeling a signal with a short time resolution.

Finally, it has the advantage of effectively applying generative adversarial training when a mel-spectrogram of low resolution is upsampled. It is known that applying generative adversarial training to the speech synthesis model helps to more realistically generate detailed representations of the target spectrogram to be generated. On the other hand, the auto-regressive model is appropriate for generating a signal that maintains long-term dependency well while being conditioned to the given time series score input. In this study, a base-spectrogram that reflects pitch and text conditions is generated by using an AR model from an input score signal. Then the second model that upsamples to a more realistic spectrogram through a model that reflects adversarial training is appropriately combined, as shown in Figure 3.7. As a result of the internal experiment, it was confirmed that when adversarial training is applied without increasing the time-resolution by super-resolution network, training proceeds unstable due to the error of the input signal and the quality of the result is lowered.

Due to these advantages, instead of modeling a full-resolution mel-spectrogram at once, we first generate a low time-resolution mel-spectrogram as an AR model and then upsample it through an adversarially trained super-resolution network. However, since the time-resolution of the input representation is not high, this model has a limitation in that it is difficult to control more detailed timing in input score information. Therefore, for better controllability, it is necessary to increase the time-resolution of the input signal and, at the same time, try to expand the system to model low-level features in the internal processing step.

3.6 Conclusion

In this chapter, we proposed the end-to-end Korean singing vocie synthesis system. We showed that using text information to model the phonetic enhancement mask actually worked, and produced more accurate pronunciation. Also, we successfully applied the conditional adversarial training method to the super-resolution stage, which resulted in a higher quality voice.

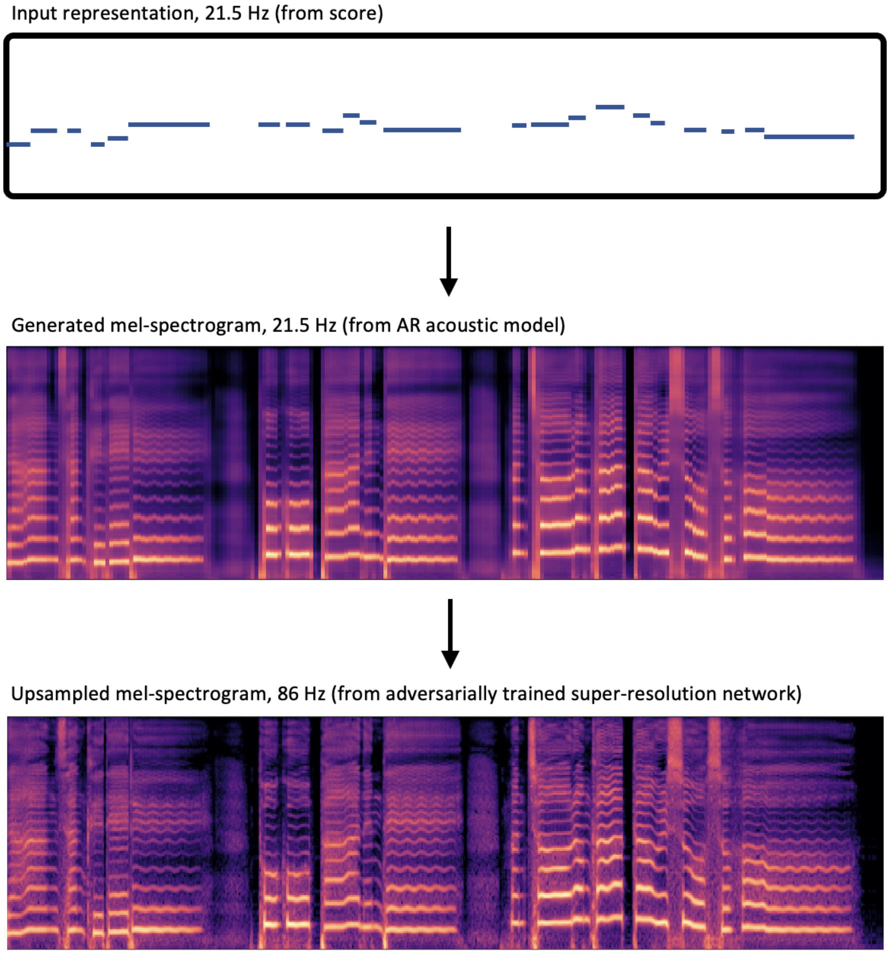

Input representation, 21.5 Hz (from score)
Generated mel-spectrogram, 21.5 Hz (from AR acoustic model)
Upsampled mel-spectrogram, 86 Hz (from adversarially trained super-resolution network)

Fig. 3.7 Illustration for the input representation (top), the output of the mel synthesis network (mid), and the output of the super-resolution network (bottom). It can be seen that smoothing is reduced, and more detailed expressions are reflected in the spectrogram after being upsampled through the super-resolution network.

Chapter 4

Disentangling Timbre and Singing Style with multi-singer Singing Synthesis System

4.1 Introduction

Singing voice synthesis (SVS) is a task that generates a natural singing voice from given sheet music and lyrics information. SVS is similar to the text-to-speech (TTS) system in terms of synthesizing natural speech from text information but differs in that it requires controllability of the duration and pitch of each syllable. Similar to the development of TTS [14, 26], the methodology based on the deep neural network has recently been studied in SVS, and the performance is comparable with the existing concatenative method [30].

After the successful development of single-singer model, researches have been conducted to extend the existing model to a multi-singer system. The multi-singer SVS system should not only produce natural pronunciation and pitch contour but also suitably reflect the identity of a particular singer. To achieve this, methods for adding conditional inputs reflecting the singer's identity to the network have been proposed [35, 43].

In this study, we break down a singer's identity into two independent factors: timbre and singing style.

A timbre is defined as a factor that allows us to distinguish the difference between the two voices even when the singers are singing with the same pitch and pronunciation, and it is generally known that they are related to singers' formant frequency [65, 66]. Meanwhile, a singing style can be defined as an expression of a singer, hence the natural realization of a pitch sequence from sheet music, including singing skills such as legato, vibrato, and so on.

The expressive SVS system should be able to synthesize the two elements effectively, and it becomes more powerful if the user can control them independently.

To this end, we propose a conditioning method that can model timbre and singing styles, respectively, while extending our existing single-singer SVS system [67] to a multi-singer system. First, we add a singer identity encoder to the baseline model to capture the singer's global identity. Then we independently condition the encoded singer identity information to the two decoders responsible for formant frequency and pitch contour so that timbre and singing style can be reflected as shown in Fig. 1. Our proposed network can independently control the two identities we define, so cross-generation combining different speakers' timbre and singing styles is also possible. Using this, we generated a singing voice that reflects the timbre or singing style of a particular singer and conducted a listening test, confirming that the network can generate a high-quality

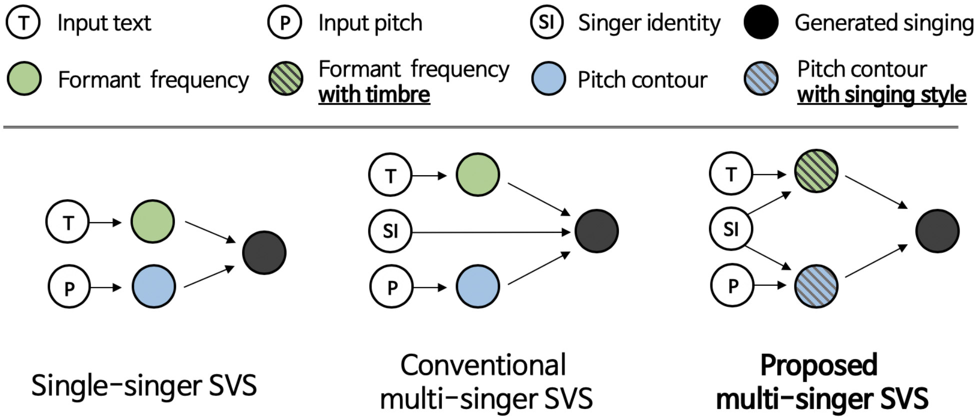


Fig. 4.1 A schematic diagram of the design of a singing voice synthesis system in which lyrics, pitch, and speaker are given. In the conventional multi-singer method, singer identity is considered as one factor, whereas in the proposed model, it is conditioned by dividing it into detailed factors related to pitch and pronunciation, respectively.

singing voice while actually reflecting each identity.

The contribution of this paper is as follows: **1)** We propose a multi-singer SVS system that produces a natural singing voice. **2)** We propose a new perspective on the identity of the singer – timbre and singing style – and propose an independent conditioning method that could model it effectively.

4.2 Related works

The concatenative method, one of the typical SVS systems such as [48, 50, 49], synthesizes the singing voice for a given query based on the pre-recorded actual singing data. This method has the advantage of high sound quality because it uses the human voice directly, but it has a limitation in that it requires an extensive data set every time a new system is designed. For a more flexible system, parametric methods have been proposed that directly predict the parameters that make up the singing voice [68, 69, 30].

This method overcomes the disadvantages of the concatenative method but has a limitation that depends on the performance of the vocoder itself. Recently, researches are being conducted to generate spectrograms using fully end-to-end methods directly [67], or designs of vocoders as trainable neural networks are also in progress [43]. In this study, we experimented based on the end-to-end network that directly generates a linear spectrogram.

4.2.1 Multi-singer SVS system

Researches to extend the SVS system to the multi-singer system has been conducted relatively recently. [35] proposes a method of expressing each singer's identity by one-hot embedding. This method is straightforward and simple, but has the limitation of requiring re-training each time to add a new singer. A method of learning trainable embedding directly from the singer's singing query for a more general singer identity is proposed in [2].

Our proposed method is different from the previous works in that it directly maps the singing query into an embedding, and defines the singer identity as two independent factors, timbre and singing style.

4.3 Proposed Method

We propose a multi-singer SVS system that can model timbre and singing styles independently. We designed the network with [67] as the baseline and extended the existing model to the multi-singer model by adding 1) singer identity encoder and 2) timbre/singing style conditioning method. As shown in Figure 4.2, our model uses text $T_{1:L}$, pitch $P_{1:L}$, mel-spectrograms $M_{0:L-1}$ of length L, and

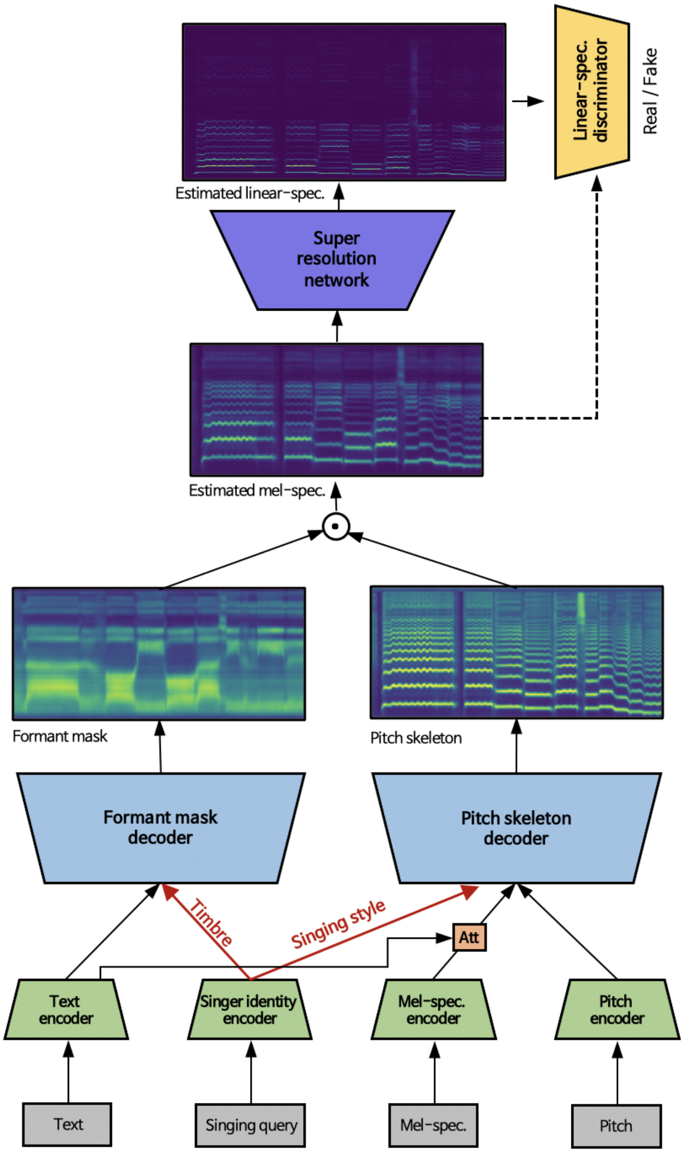


Fig. 4.2 The overview of the proposed multi-singer SVS model that can model the singer's identity by dividing it into two independent elements, singing styke and timbre

a singing voice query Q as inputs. Each input is encoded via an encoder, then are decoded with formant mask decoder and pitch skeleton decoder. The formant mask decoder generates a pronunciation and timbre-related feature FM from encoded text E_T and query E_Q. The pitch skeleton decoder generates pitch and style-related feature PS from encoded mel-spectrogram E_M, pitch E_P and query E_Q. Estimated mel-spectrograms $\hat{M}_{1:L}$; the result of element-wise multiplication of FM and PS, are converted to estimated linear spectrograms $\hat{S}_{1:L'}$ via a super-resolution network SR. Finally, to create a linear spectrogram that is more realistic, we applied adversarial training and added a discriminator to this end. Please refer to [67] for more detailed information on each module of the network. The summary of the generation process of the entire network is as follows:

$$\hat{S} = SR(\hat{M}) = SR(FM(T, Q) \odot PS(M, P, Q)). \tag{4.1}$$

4.3.1 Singer identity encoder

Expanding the single-singer model to the multi-singer model requires an additional input about singer identity information.

To achieve this, we designed a singer identity encoder that directly maps the singer's singing voice into an embedding vector. Then, the pooled embedding is converted into a 256-dimensional embedding vector through the dense layer and tiled to match the number of time frames of the features. Finally, it is used as a conditioning embedding vector for a pitch skeleton decoder and a formant mask decoder, respectively.

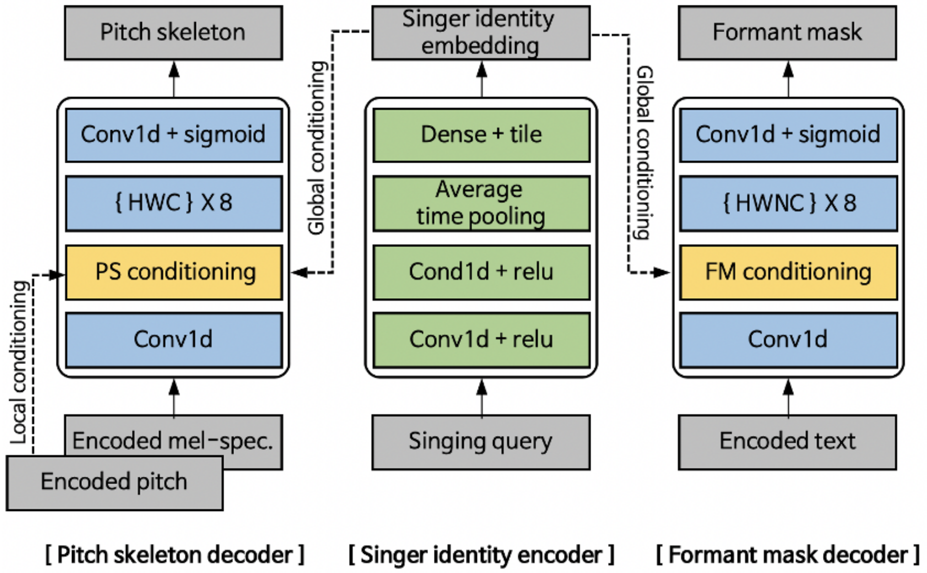


Fig. 4.3 Singer identity encoder structure and conditioning method. HWC, HWNC denotes highwav causal/non-causal covolutional module proposed in [1], and **Conv1d**, **Dense**, **relu**, **sigmoid** denotes 1d-convolutional layer, fully connected layer, rectifier linear unit, and sigmoid activation unit, respectively.

4.3.2 Disentangling timbre & singing style

In this section, we will provide details of our conditioning method to model timbre and singing styles separately. Our baseline network generates a mel-spectrogram by the multiplication of two different features, formant mask and pitch skeleton. Formant mask is responsible for regulating formant frequency to model corresponding pronunciation information from the input text, while pitch skeleton plays a role in creating natural pitch contours from input pitch. We focused that singer identity embedding could be reflected in each of these features in different ways. In other words, we assumed that singer identity embedding had to be conditioned on the formant mask decoder to control the modality of the timbre, and to control the singing style, it had to be conditioned on the pitch skeleton decoder that forms the shape of the pitch contour. Based on this assumption, we used a method of conditioning singer identity embedding independently of each of the two decoders. We used the global conditioning method proposed in [12], and the specific formula is as follows.

$$\mathbf{z}(\mathbf{x}, \mathbf{c}) = \sigma(W_1 * \mathbf{x} + V_1 * \mathbf{c}) \odot relu(W_2 * \mathbf{x} + V_2 * \mathbf{c}) \tag{4.2}$$

where $\mathbf{x}$ is a target to be conditioned, $\mathbf{c}$ is a condition vector, and $W_* * \mathbf{x}$ and $V_* * \mathbf{c}$ are 1d-convolution operations.

4.4 Experiment

4.4.1 Dataset and preprocessing

For training, we use 255 songs of a singing voice, consisting of a total of 15 singers. Three inputs (text, pitch, and mel-spectrogram) were extracted from the lyrics text, midi, and audio data, respectively. Query singing voice for singer identity embedding was randomly chosen from other singing sources of that singer. One of each singer's recorded songs was used as test data, and the rest were used to train the network.

We preprocessed the training data in the same way as [67] for all input features except the singing query for singer identity embedding. The sampling rate was set to 22,050Hz. The preprocessing step for the singing query is as follows. First, we randomly selected about the 12-second section from the singer's singing voice source. Then, we set both the window size and hop length to 1024 and converted the singing voice waveform into a mel-spectrogram of 80 dimensions and 256 frames and used it as the singing query.

4.4.2 Training & inference

We trained the network in the same way as proposed by [67], except to set different speaker samples evenly distributed in each mini-batch. The inference was also conducted in the same way as in the previous study, but for tests to show that the timbre and singing style can be controlled separately, we generated test samples through cross-generation, which generates a pitch skeleton and a formant mask from different speaker embeddings, respectively [1].

4.4.3 Analysis on generated spectrogram

We compared the generated spectrogram by a different speaker for the same pitch and text to see the effect of the speaker identity embedding. As shown in Figure 4.4, each spectrogram has a similar overall shape but includes partial

[1] audio samples available at `https://juheo.github.io/DTS`

differences. In the case of a formant mask, female vocals have vigorous intensity in high-frequency areas, while the male's corresponding frequency area is shifting down. This is in line with the fact that males generally have lower formant frequency even in the same-pitched condition. Even with the same gender, we can see that the shape of the formant mask is different, and from this, we have confirmed that the speaker embedding appropriately reflects the timbre of each singer. Likewise, pitch skeleton differs depending on the speaker, where it is spotted at the position of the onset/offset, the slope near it, the intensity of vibrato, and the shape of the unvoiced area.

From this, we confirm that the singer identity embedding affects the style change of pitch skeleton effectively. Note that despite conditioning with identical embeddings through time, changes in the style of pitch skeletons over time have been observed. We argue that our network generates singing voice in an auto-regressive way so that it could reflect the style differences over the time axis of different singers.

We were also able to observe a few changes as we interpolate two different singer identity embeddings from female to male vocalist. For example, we found that the high-frequency area of the formant mask was gradually lowered, and the vibrato was gradually strengthened in the case of pitch skeleton. From this, we confirmed that speaker embedding not only reflects the identity of different singers but also contains appropriate information about their changes.

4.4.4 Listening test

We conducted a listening test with a total of 6 different male and female singer's voices for qualitative evaluation. We generated two vocal voices for each per-

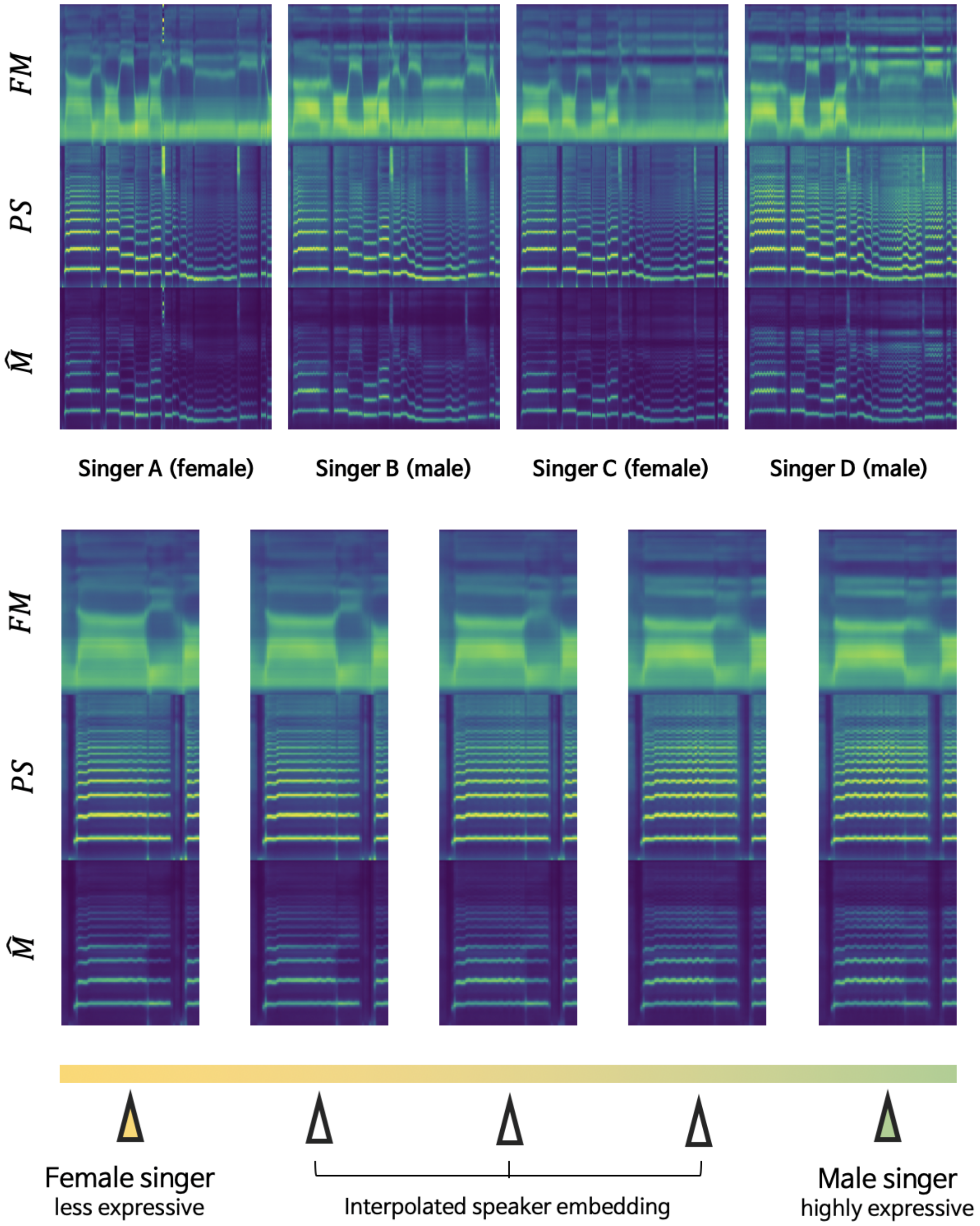


Fig. 4.4 Generated mel-spectrogram with various singer embedding (top) and interpolated singer embedding (bottom). FM, PS, $\hat{M}$ denotes formant mask, pitch skeleton, and estimated mel-spectrogram, respectively.

Table 4.1 Listening evaluation result according to whether cross generation is applied (9-point scale). There was no statistically significant difference depending on whether or not it was applied, so it was confirmed that the cross-generation method could be used without performance degradation.

Model	**Pronun.acc**	**Sound.quality**	**Naturalness**
proposed (w/o cross)	7.30 ± 1.44	5.06 ± 1.44	5.64 ± 2.01
proposed (w/ cross)	7.36 ± 1.39	5.19 ± 1.76	5.55 ± 2.02
Ground	7.43 ± 1.50	6.40 ± 1.96	6.89 ± 1.89

son for the randomly selected song. To show that the proposed network does not have any degradation in performance even when it independently controls singing style and timbre, we also created two samples for each person's formant mask with another person's pitch skeleton and used them for evaluation. 26 participants were asked to evaluate pronunciation accuracy, sound quality, and the naturalness of test samples on a 9-point scale ranging from very bad to very good. The result is shown in Table 4.1.

A paired t-test [70] shows no significant difference for all items, regardless of whether the cross-generation was carried out. We also confirmed that there is no significant difference with the ground truth samples for the pronunciation accuracy. From this, we verify that our proposed network could combine different timbre and singing style without any performance degradation, and can generate a singing voice that can match the ground truth sample with accurate pronunciation.

4.4.5 Timbre & style classification test

We conducted a classification test to ensure that the network generates singing voice that reflect timbre and singing styles independently. We prepared a total of 20 test sets, 10 each for judging timbre and singing style, and each test set

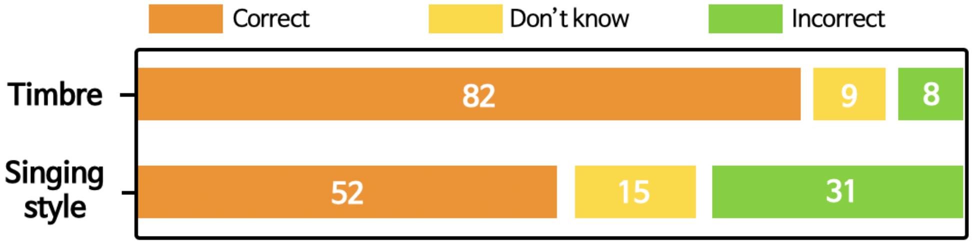


Fig. 4.5 Timbre and style classification test result

consisted of three sources A, B, and C. A and B are the singing voices generated without cross-generation, and C is cross-generated using its own timbre/style and referencing one of A or B’s style/timbre. By comparing these samples, participants are asked to prefer instead sample C’s timbre/style is a closer match to A or B’s. Considering gender differences, we equally divided three singers’ gender into every possible combination, and the result is as follows in Figure 4.5.

According to the results of the experiment, 8% of participants in timbre and 31% in singing style chose incorrect answers. The answer rate of the singing style was lower than the timbre, which is analyzed because the data we used in training consisted of amateur vocals whose style was relatively unclear. Nevertheless, more than half of the participants responded to the correct answer from which we conjecture that our network is able to generate a timbre and singing style that matches a given singer identity query to a level that humans can perceive.

4.5 Discussion

4.5.1 Query audio selection strategy for singer identity encoder

Our proposed singer identity encoder encodes the singer's characteristics from the mel-spectrogram of a part of the audio section sung by the singer. This method has an advantage over using a fixed method such as one-hot encoding in that it is possible to extract singer characteristics from arbitrary audio. However, since the model is trained with the goal of generating a specific singing section again from the extracted singer embedding information, it is important to determine a strategy for which query to select.

The query selection strategy can be summarized in three ways, as shown in Figure 4.6. The first is to use the audio itself to be generated during training as a query. The second is a method of randomly selecting among all the songs of the singer, such as the audio to be generated during training. The third is a method of randomly selecting among the sections near the audio section to be generated during training. We tested three methods, and as a result, we confirmed that the third strategy is the best method.

The first method is a strategy that gives the most information from the model's point of view that needs to reconstruct the target audio. However, this method has a disadvantage in that the controllability is lowered. To generate target audio, the model must properly encode pitch, pronunciation, and speaker information. However, if the same audio as target audio is used as the query, pitch and pronunciation information also tends to be inferred from the query audio. We found a phenomenon in which information is counted, so when generating with a trained model later, it tries to generate a sound source similar to the query audio without faithfully generating the input score. Therefore, us-

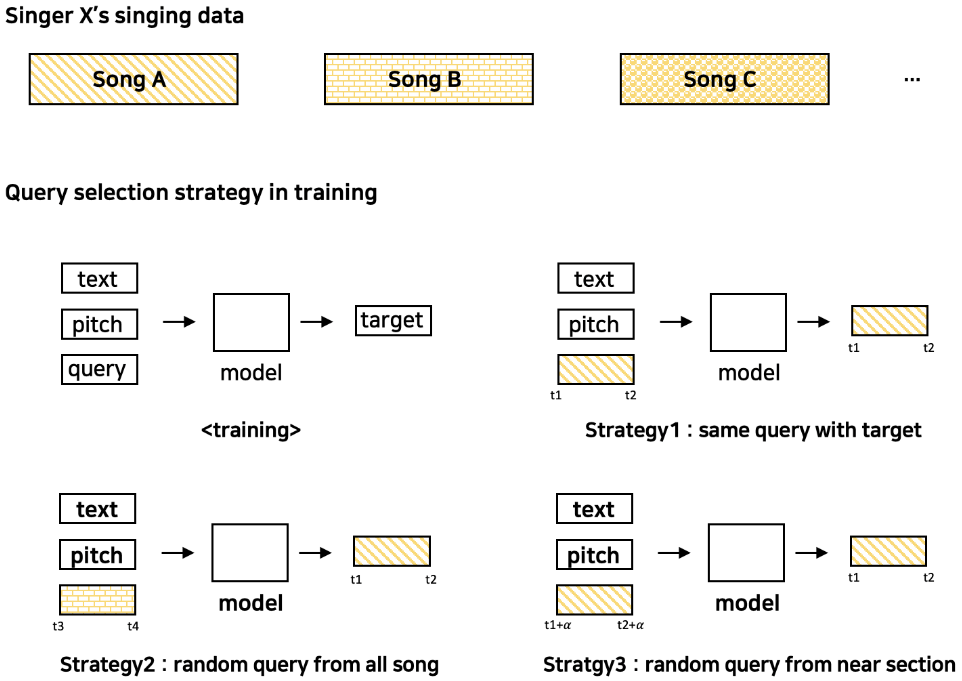

Singer X's singing data
Song A
Song B
Song C
...
Query selection strategy in training
text
pitch
query
model
target
<training>
text
pitch
model
t1
t2
t1
t2
Strategy1 : same query with target
text
pitch
model
t3
t4
t1
t2
Strategy2 : random query from all song
text
pitch
model
t1+α
t2+α
t1
t2
Stratgy3 : random query from near section

Fig. 4.6 A schematic diagram of a strategy for selecting a query to input into the singer identity encoder. The number below the square where the query is represented means the start and end times of the section selected in the song. The query can be selected the same as the target audio or randomly selected all songs by the same speaker. However, in either case, it is not suitable for modeling. Therefore, we used strategy 3, which randomly selects the query audio among the sections near the target audio.

ing the target audio as a direct query is not appropriate because it does not sufficiently disentangle the speaker and content information.

The second method is to select a random section of all songs by the same singer as target audio as a query. This method is appropriate from the viewpoint of disentanglement because target audio and query audio have the same singer information and different content information. However, this strategy has the disadvantage of smoothing various characteristics of the singer. Singers generally use different timbre and singing styles depending on the song and section. Therefore, the singer identity encoder is trained in a way that encodes only the average voice without richly reflecting the singer's characteristics. If an arbitrary query not related to the target audio is used in the training process, the effect of averaging the features of subtly different singers as one central feature occurs. Therefore, this method is not suitable for higher expressive power.

Therefore, we adopted the third method. The third method uses the nearby audio in the target audio section as a query. This method has the advantage of containing a similar timbre and singing style because the target audio and contents are different because they are nearby sections. We randomly selected a 10-second interval among the surrounding 20-second intervals of target audio and used it as query audio. As a result, it was confirmed that the singer identity encoder is trained in a direction that reflects the various characteristics of the singer that are not related to the contents.

4.5.2 Few-shot adaptation

Our proposed multi-speaker singing synthesis model can reflect the singing style and timbre of singers included in the training data well. Furthermore, we internally conducted a few-shot adaptation experiment for an unseen singer who was not included in the training and devised a method that can clone the speaker's voice and timbre with little data. First, our proposed model has scalability for any speaker because the speaker embedding is not fixed and is trained through the speaker identity encoder. Therefore, we can perform note-level annotation on a new speaker and then use the data to go through the same process as pre-training to perform adaptation for the new speaker.

Specifically, the following methods were used for few-shot adaptation training. First, as is well known through various previous studies [71, 72], when transfer learning the pre-trained model, training was carried out with the learning rate reduced 10 times lower compared to the previous one. As a result of the comparative experiment, it was confirmed that the transfer of timbre and singing style occurred while maintaining pronunciation and pitch more stably compared to the case where the learning rate was not reduced. We tried to train only the decoder part while freezing modules such as text encoder and pitch encoder unrelated to speaker information. However, it showed the best performance when all the model parameters were trained. We also applied adversarial training in the transfer learning process. At this time, as suggested in [73], we tried to freeze the upper layer of the discriminator for efficient transfer learning without discriminator over-fitting and catastrophic forgetting, but there was no significant performance difference. In addition, we attempted to train together with the existing training data to avoid losing the expressive power of the previously trained pronunciation and pitch. There was a trade-off between the similarity of timbre and singing style with the target speaker and proper pronunciation. Finally, we found that the model overfits a small amount of data when iteration is repeated a lot. The early stopping technique for validation loss was applied to select the endpoint of transfer learning.

As a result of the few-shot speaker adaptation training, as shown in the Figure 4.7, it was confirmed that the results reflected not only the timbre but also the singing style of each unseen singer. The reason that transfer learning was possible with only a small amount of data corresponding to about 3 minutes is thought to be because the proposed model is a system that independently separates and models pronunciation, pitch, timbre, and singing style.

However, adapting to a new singer has a limitation in that note-level annotation process on the singing voice of the corresponding speaker is required. In fact, as a result of the internal experiment, it was confirmed that when the voice of an unseen singer was put into the speaker identity without transfer learning to generate a singing, the result was generated with the most similar voice seen during the training process. In other words, our proposed model is not sufficiently generalized for the voices of various singers. This problem can be solved by expanding the size of the dataset, but research in a different direction is needed because it is difficult to secure enough singing data. We can secure data by creating pseudo labels for unlabeled data using a note-level transcriber or phoneme recognition network. After that, we pre-train the SVS model for large-capacity singing sources based on this. We will then be able to secure generalization ability for various voices.

4.6 Conclusion

In this chapter, we proposed a multi-singer SVS system that can independently model and control the singer's timbre and singing style. We disentangled the

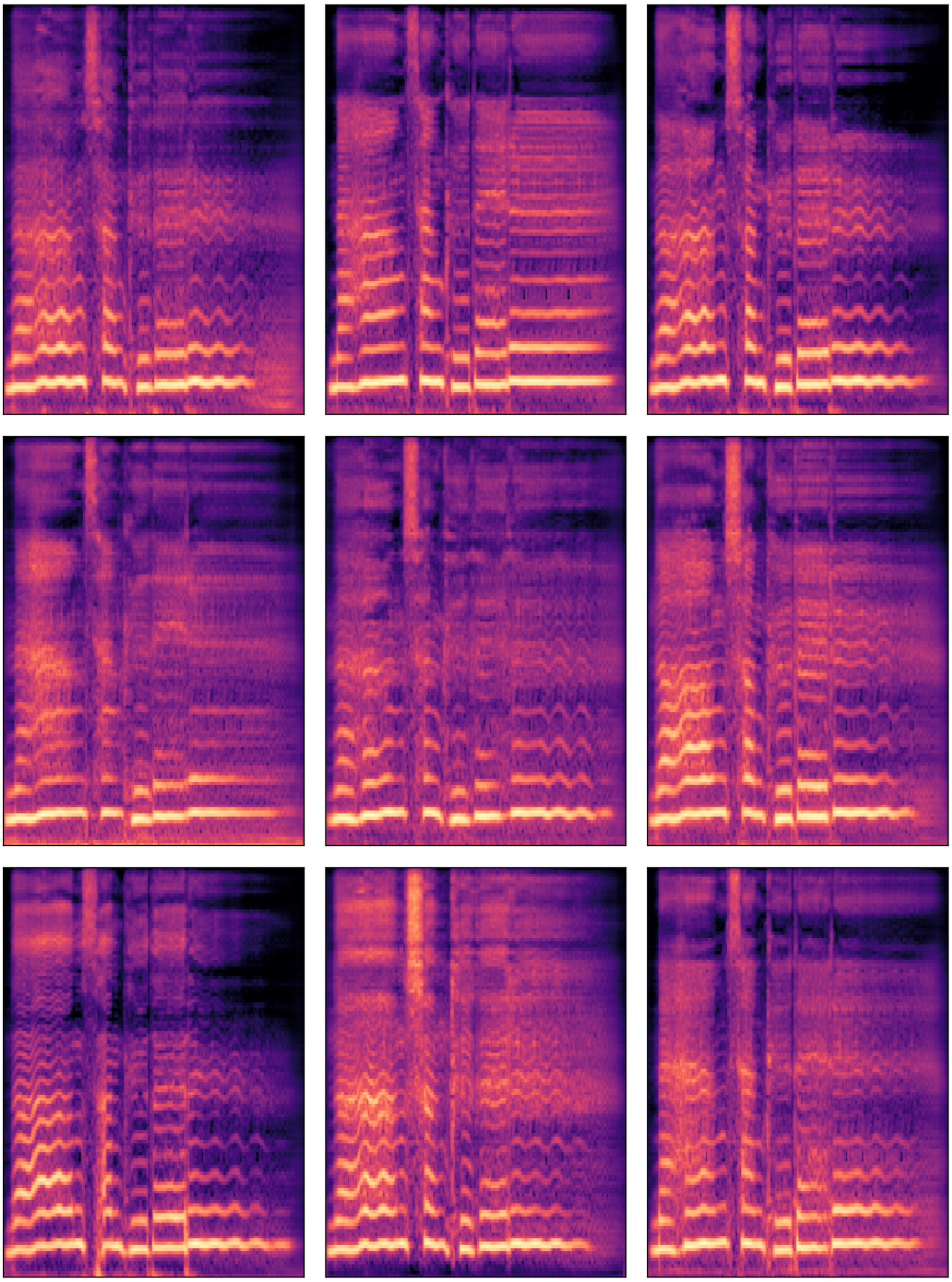


Fig. 4.7 A spectrogram of the result of generating the same score input with 9 different unseen singer identities that performed few-shot adaptation with one training song. We can confirm that the timbre and pitch contour expression methods differ depending on the speaker.

identity of the singer through a method of conforming singer identity embedding independently in two decoders. The listening test showed that our system could produce high quality and accurate singing comparable to the ground truth singing voice. Through listening tests, which classify the timbre and singing styles of the generated samples, we revealed that we could control both elements independently.

Chapter 5

Expressive Singing Synthesis Using Local Style Token and Dual-path Pitch Encoder

5.1 Introduction

Singing voice synthesis (SVS) is the task of generating a natural singing voice from a given musical score. With the development of various deep generative models, research on synthesizing high-quality singing voice has been emerging recently [37, 74, 75, 39]. As the performance of the SVS improves, there are increasing cases in which the technology is applied to the production of actual music content [76]. singing is drawing attention. Accordingly, the SVS system that can control various musical expressions by reflecting the user's intention is drawing more attention.

There are two challenging problems in building a SVS system that can easily control various expressions. The first is that it is difficult to build datasets in which various musical expressions are annotated. Unlike information such as pitch and lyrics, which are relatively easy to label, expressive elements such as breathing, intensity, and singing techniques related to pitch control are more expensive and time-consuming to label.

The second problem is that the more information the user has to initially enter into the SVS system, the more burdensome it will become to the user. In general, the SVS system takes a MIDI pitch sequence and lyrics as an input. Although lots of input parameters such as musical expressions (e.g., breath, intensity, vibrato parameters) can be given to the system, it is inconvenient for the user because the amount of input parameters one has to specify in frame-level increases whenever creating a new song.

Therefore, to build an SVS system that is easy to use and can control various expressions while dealing with these problems, 1) expression elements should be trained based on self-supervised manners, and 2) the parameters for the detail expressions should be generated automatically during the initial generation stage, but should still be able to be modified and resynthesized if desired.

To this end, we propose an SVS capable of controlling expression along with two novel methods. First, to model a variety of unlabeled style representations, we introduce a Local Style Token (LST) module that captures styles in a self-supervised manner from given text and pitch information based on [16]. Unlike [16], however, we do not design the model to infer a single global style vector from a reference signal, but rather predict frame-wise style tokens that change over time. Second, in order to take control over f0 contour, we introduce a Dual-path Pitch Encoder (DPE) that is able to selectively use MIDI pitch and f0 contour as an input. In the training process, the output of the two encoders is randomly selected to produce the same result. In the generation process, we can resynthesize the singing by freely controlling the f0 contour extracted from the results generated by the MIDI pitch.

Through the quantitative and qualitative evaluation, we confirmed that the proposed system allows free control of expressions such as breathing, intensity, and detailed f0 techniques while producing a high-quality singing voice.

The main contributions of this study are as follows:

- We propose a content-driven local style token module that can model various musical expressions in singing such as intensity and breathing, trained in a self-supervised manner.
- We propose a dual-path pitch encoder that takes either MIDI pitch sequence or f0 contour, allowing the users to control pitch both at a coarse or fine level at one's choice.

5.2 Related work

Recently, interest in research on the SVS system that can reflect musical expression is increasing. A method of explicitly modeling information such as pitch curves, energy, V/UV, etc., which can be extracted directly from the vocal signal, was proposed in [74]. [75] proposed a method to interpret the music score more naturally by introducing a module that predicts the difference between the actual singing and the score. Efforts to create natural pitch contour have also been made in various ways, such as directly predicting f0 from note sequences [77, 78, 79], or predicting variables of the parametric f0 contours [80].

Despite various kinds of efforts to improve the expressive power of the SVS system, there has been no study yet that, to our knowledge, allows users' to control singing style elements that cannot be extracted directly from the signal.

5.3 Proposed method

Our proposed SVS model is designed by adding a local style token (LST) and dual-path pitch encoder (DPE) to model and control various expressions based on [81]. Based on the source-filter theory, the acoustic model of our SVS system is an auto-regressive model including two decoders that generate the filter and the source signal, respectively. The filter and source were designed to be modeled from (text) and (pitch, previous acoustic feature), respectively, where singer embedding and LST are used together as conditions. Specifically, acoustic feature $\hat{M}$ is generated as follows:

$$\hat{M} = D_F(E_t, cat(E_s, T_t)) + D_S(E_m, E_p, cat(E_s, T_p)), \tag{5.1}$$

where D, T, E, and subscripts F, S, t, p, s, m denote decoder, local style token sequence, and encoder output, filter, source, text, pitch, singer, and acoustic feature, respectively. Finally, the generated acoustic feature $\hat{M}$ is converted to a waveform through a vocoder. The entire network is trained with acoustic feature reconstruction loss and adversarial loss, and the overview of the acoustic model structure is shown in Figure [**?**].

5.3.1 Local style token module

To model unspecified singing expressions in a music score, we introduce a local style token module. We assume that musical expression elements in singing

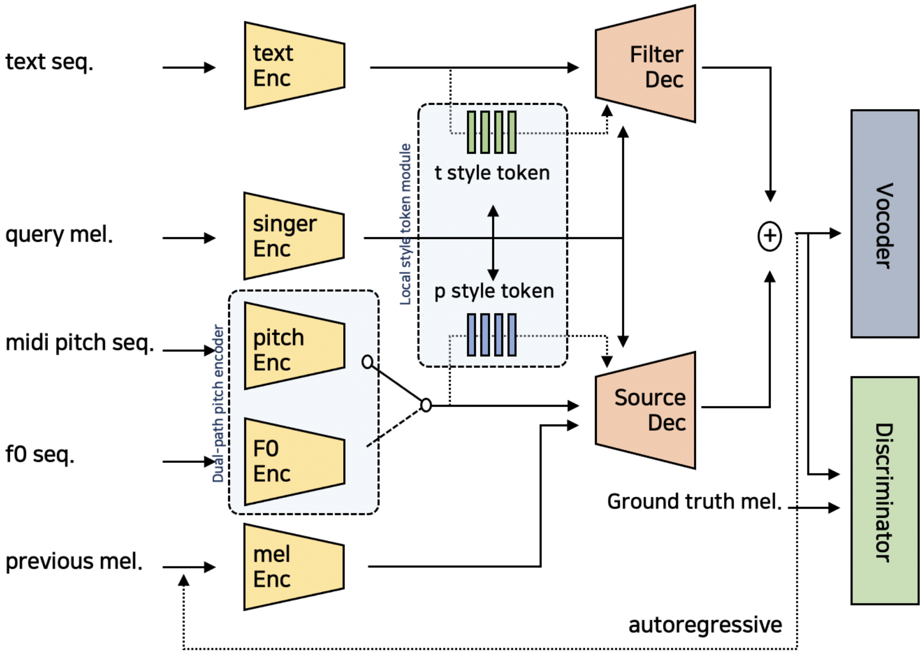


Fig. 5.1 Schematic diagram of proposed expressive SVS system with local style token module and dual-path pitch encoder.

can be inferred from a given input text and pitch sequence and that these elements should exist in a time-varying form. To achieve this goal, we modify the attention mechanism proposed in [16] and retrieve a local style token sequence by referencing the input contents such as pitch and text. We first introduce a style encoder consisting of stacked 1d-CNN layers with gated linear units [82] to obtain query sequence $Q_t, Q_p \in R^{L \times d}$ from text E_t, pitch E_p and singer E_s embedding sequences as follows:

$$Q_{t/p} = \text{StyleEnc}(cat(E_{t/p}, E_s)) \tag{5.2}$$

where L, d denotes sequence length and channel dimension, respectively. Because the LST module operates exactly the same on both text t and pitch p sides, we denote the subscript t or p as t/p through out the paper.

Then, N randomly initialized trainable style key and value $K_{t/p}, V_{t/p} \in R^{N \times d}$ is used to obtain the style score $S_{t/p} \in R^{L \times N}$, and is computed as follows:

$$S_{t/p} = \text{softmax}(\frac{Q_{t/p} K_{t/p}^T}{\sqrt{d}}) \tag{5.3}$$

Finally, we obtain a LST sequence via matrix multiplication between $S_{t/p}$ and $V_{t/p}$ as follows: $T_{t/p} = S_{t/p} V_{t/p}$. In the inference stage, the predicted LST sequence from the input contents may be used as it is, or the style score $S_{t/p}$ can be modified in the desired way to control the musical expression of the singing voice as shown in Fig. 5.3. The overview of the LST module is illustrated in Figure 5.2.

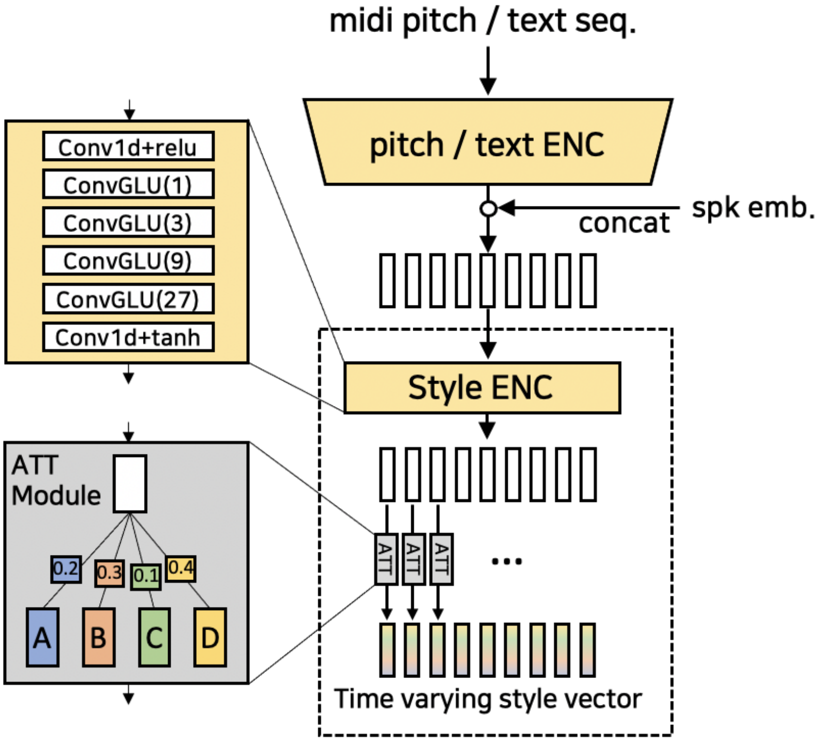


Fig. 5.2 The proposed local style token module diagram. This module models time-varing expression elements by performing attention operations with pre-defined style vectors in a self-supervised manner from the input musical score pitch or lyric information. $ConvGLU(k)$ denotes 1D-CNN layer with gated linear unit in which kernel size is k.

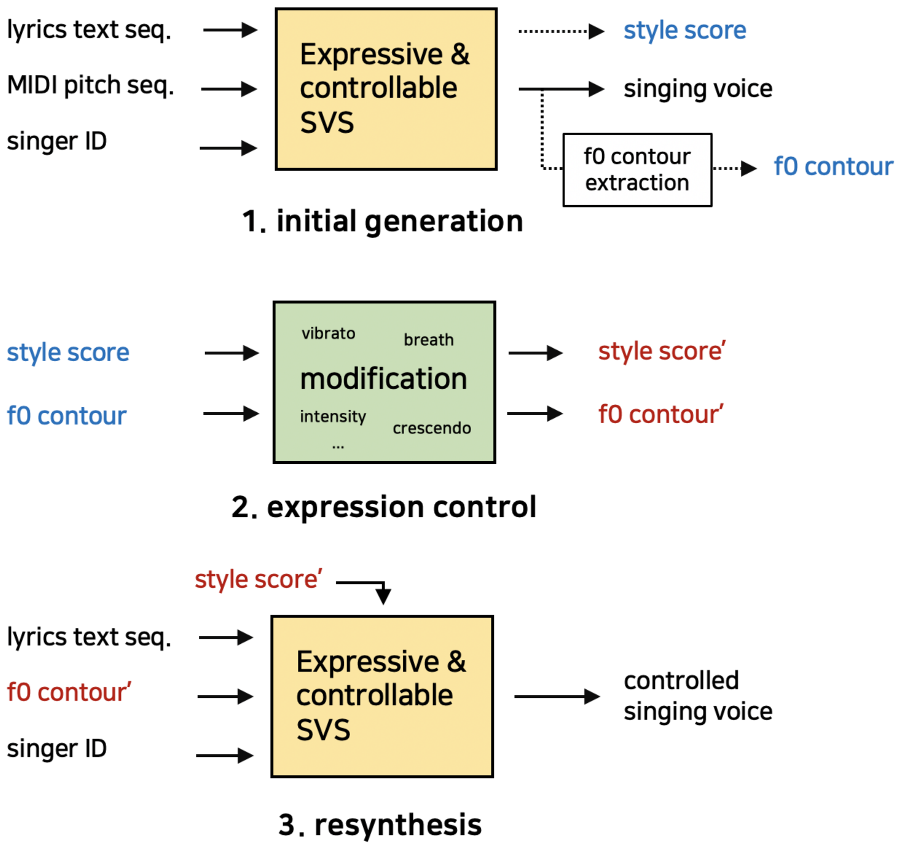


Fig. 5.3 The control procedure of the proposed SVS system. First, the singing voice is generated using the music score as an initial generation step. We obtain style score and f0 contour from the initially generated singing voice. Next, the style score and f0 contour is modified as desired by the user. Finally, we obtain the desired singing voice by resynthesizing it with the modified inputs.

5.3.2 Dual-path pitch encoder

Another important factor that determines the expressiveness of singing is the f0 contour. Modeling a natural f0 contour from a MIDI pitch sequence is one of the important research areas of SVS research [78, 77, 83]. However, the natural f0 contour must be carefully determined by referencing not only MIDI pitch information, but also text, singer, and context, etc [79, 84].

Meanwhile, the model proposed in [81] produces a spectrogram having natural f0 implicitly reflecting information from inputs to the system such as singers, MIDI pitch sequence, and lyrics. Inspired by this, we aim to design a model that can use both MIDI pitch and f0 contour as inputs instead of making an additional model that predicts f0 contour explicitly.

To this end, a dual-path pitch encoder with the same structure in which two inputs of pitch and f0 contour can be freely used is proposed, and the training is conducted by randomly selecting one of the two pitch representations. This way, we can generate singing voices with natural f0 contour from the initial generation using MIDI pitch input. If we want to further control pitch techniques such as vibrato or portamento, we can modify f0 directly and recreate them using f0 encoder as shown in Figure 5.3.

5.3.3 Bandwidth extension vocoder

We used a HiFiGAN vocoder [85] to convert the generated acoustic feature into a waveform. Interestingly, we found that the HiFiGAN vocoder can perform both bandwidth extension and waveform generation simultaneously. That is, we trained the vocoder to convert a 22.05khz acoustic feature generated by the acoustic model into a 44.1khz waveform to generate a higher quality sound source without having to train an acoustic model that generates 44.1khz acoustic feature.

5.4 Experiment

5.4.1 Dataset

We used 1,150 singing voices of 88 females and 66 males for training. Each singing voice is paired with manually annotated music scores. The sampling rate of the singing voices was set to 44.1khz, and the notes are annotated for each syllable. A phoneme-level annotation of lyrics was done by assigning one frame to onset and coda each, and the rest of the frame to vowel as proposed in [67].

5.4.2 Training

The training of the models was done the same as in [81] except for the newly proposed LST and DPE. The number of the style tokens N was set to 4, and training was conducted using either MIDI pitch sequence or f0 contour with a 50% probability for each pitch input. We used WORLD [64] to extract f0 contour from the singing voices. The encoder and decoder structure of the model is the same as [81] except that all highway convolutional units have been changed to GLUs. The structure of the style encoder is as shown in Fig. 1-(b), and the structure of the f0 encoder is the same as that of the MIDI pitch encoder except for the input channel of the first 1d-CNN layer. We used a 128-dimensional mel spectrogram extracted from the waveform of the 22.05kHz sampling rate with window size and hop length set to 1024, 256, respectively, as an acoustic feature. The acoustic model was trained using the adversarial loss proposed by [67] along with L1 loss.

We trained a total of three models for comparative experiments to see if LST and DPE help improve controllability while generating high-quality singing voices. The three models are as follows: 1) **Single** model has only MIDI pitch encoder without an f0 encoder. The f0 contour is controlled by modifying the initially generated singing voice with WORLD vocoder [64]. 2) **Dual** has a DPE including both an f0 and a MIDI pitch sequence encoder. 3) **DualLST** model incorporates both the DPE and LST modules.

5.4.3 Qualitative evaluation

To examine the generation quality and controllability of the proposed model, we organized two test sets and for listening evaluations. The first test set consisted of singing voices generated with MIDI pitch and text input. A total of 10 male and 10 female singers were randomly selected to create 10 musical verses each. To account for degradation from vocoders, we resynthesized the waveforms from ground truth acoustic features using the vocoders, and included them in the listening evaluations. Secondly, to verify if it is possible to control the f0 contour extracted from the initially generated results, we conducted a pitch shift experiment. For this we generated 60 audio pairs, each pair generated with 3 different f0 contours, that is, initial f0, +2 semitone shift, and -2 semitone shift. This results in 180 audio samples in total.

The listening evaluations was conducted through Amazon Mechanical Turk, and the overall naturalness and pitch naturalness of each sound source were evaluated. The evaluation results for each test are shown in Table 5.1 and

Table 5.2, respectively. [1]

Table 5.1 Initial generation test MOS result

Model	**overall naturalness**	**pitch naturalness**
Single	3.86 ± 0.05	3.89 ± 0.05
Dual	3.79 ± 0.06	3.86 ± 0.06
DualLST	3.84 ± 0.05	3.87 ± 0.05
Recon	3.89 ± 0.05	3.90 ± 0.06
GT	4.03 ± 0.05	4.06 ± 0.05

Table 5.2 Pitch shift test MOS result

Model	**-2**	**0**	**+2**
Single	3.75 ± 0.18	3.69 ± 0.18	3.63 ± 0.22
Dual	3.80 ± 0.16	3.82 ± 0.17	3.80 ± 0.19
DualLST	3.80 ± 0.18	3.92 ± 0.15	3.78 ± 0.16

From the results in Table 5.1, we confirmed that all of the models obtained naturalness results that did not differ significantly from reconstructed singing voices by HiFiGAN vocoder. Using DPE resulted in a slightly lower naturalness score than the Single model, which seems to be the result of the ambiguity that occurred in the process of producing the same result from two different types of inputs.

Note that the Dual model can take control over f0 contours with a negligible performance drop in overall naturalness.

Table 5.2 shows that using DPE can still produce high-quality sound sources even when the pitch is shifted than using the parametric vocoder capable of f0 control.

In all cases, the Dual and DualLST models showed better naturalness than the Single model. Although we performed the global pitch shift in the exper-

[1] Audio sample : `https://tinyurl.com/cpyfbt6h`

iment to maintain the temporal context of the song, the pitch shift can be applied to local sections. That is, the model we proposed can naturally reflect various pitch techniques such as vibrato, attack, and release, as introduced in 3.5.

5.4.4 Dual-path reconstruction analysis

A reconstruction analysis was performed to quantitatively confirm if DPE helps faithfully reflect the input pitch information. First, we initially generated 150 audio samples from randomly selected singers and phrases. Then, we extracted f0 contours from the generated samples and the reconstructed them using the extracted f0 contours. Finally, we measured Mel Cepstral Distortion (MCD), f0-RMSE, and V/UV error rate between the initially generated sample and the reconstructed sample. The results in Table 5.3 show that the difference between the initially generated sample and the reconstructed sample is negligible showing that the proposed DPE module is working as we intended. As a reference, we also report the error between the ground truth audio sample and reconstructed sample using a HiFiGAN vocoder, which is shown as Recon in Table 5.3. This shows that the difference between the initially generated sample and the reconstructed sample is sufficiently small even when compared to the difference between the ground truth audio sample and reconstructed sample using the vocoder. In particular, adding the LST module always helped lowering evaluation measures, indicating that the LST module not only helps control singing expressions but is also helpful for generating better output with accurate f0 contour.

Table 5.3 MCD, f0-RMSE and V/UV for reconstruction test

Model	**MCD(dB)**	**f0 RMSE(Hz)**	**VUV(%)**
Dual	3.03	7.21	3.04
DualLST	2.95	7.06	2.93
Recon	4.69	5.61	3.54

5.4.5 Qualitative analysis

To qualitatively examine the controllability of the proposed methods, we tried various style modifications by manipulating the initial LST sequence and f0 contour [2].

Breath control | The first noticeable style captured by the style tokens was the token activated in the breathing section of the phrases. We named it as breath token ($V_b \in V_t$). By scaling the style score ($S_b \in S_t$) of the breath token by 0.5-2 times, we found that we can easily control the intensity of breathing sounds as shown in Figure 5.4-bottom.

Intensity control | We also found that one of the tokens captures the intensity of the singing voice. We named it as an intensity token ($V_i \in V_t$). As shown in Figure 5.4-top, we found that we can change the intensity of the singing voice, which is shown explicitly by an energy contour. Taking advantage of this, we found that we can control musical expressions such as crescendo or decrescendo by linearly increasing or decreasing the attention score of the intensity token ($S_i \in S_t$). Note that this control is different from simply increasing or decreasing the volume of the waveform.

[2] The role of each of the N style tokens is randomly permuted for every experiment. In addition, although breath and intensity were always captured by the style tokens in the text side (T_t), no meaningful token was found in the pitch side (T_p).

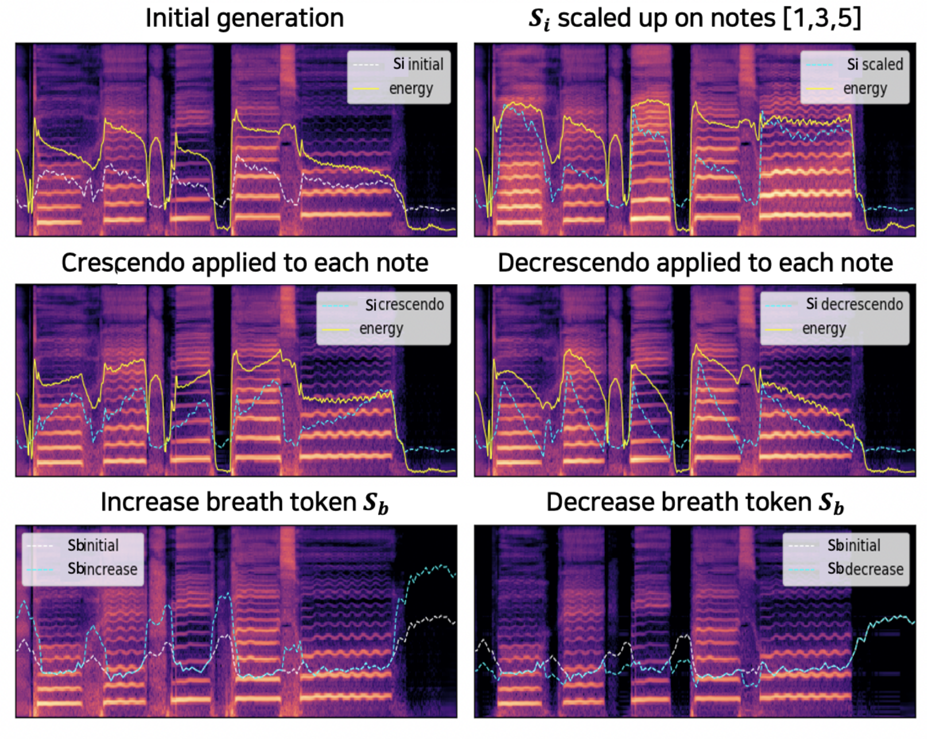


Fig. 5.4 Spectrogram analysis with intensity, breath control applied with local style token module. If the value of intensity token is changed, it can be confirmed that the energy of the generated result changes together. If control the value of the breath token, it can be confirmed that the volume of the breath changes.

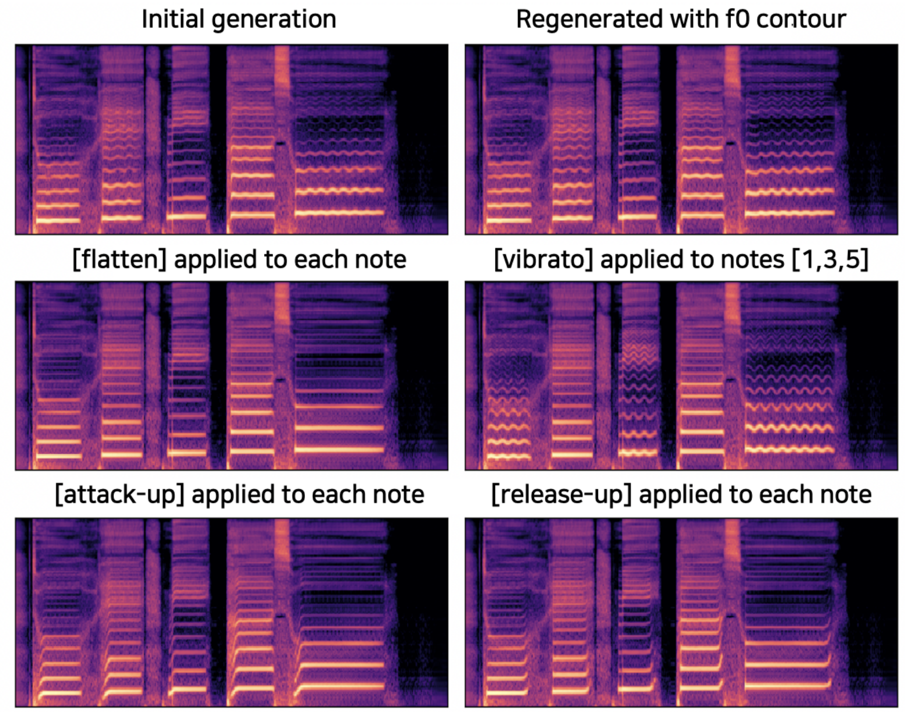


Fig. 5.5 Spectrogram analysis with f0 control applied with dual-path pitch encoder . It can be seen that various modification such as flatten, vibrato, attack, and release can be applied to the f0 contour of the initially generated result.

f0 contour control | We can easily control f0 contour using DPE with simple operations such as reducing variance (flatten), adding sinusoidal values (vibrato), adding or subtracting small values from the onset and offset positions of the note (attack/release up/down), as shown in Figure 5.5.

5.5 Discussion

5.5.1 Difference between midi pitch and f0

Our proposed model has a dual-path pitch encoder structure that accepts both midi pitch and f0 as inputs. Using this, users can first create an initial singing voice based on the input of the midi note sequence, then modify and recreate the desired f0 contour of the generated result. It may be possible to reflect the desired detailed correction to the midi pitch input without a module to process the f0 input. However, we adopted the dual-path pitch encoder structure for the following reasons.

First, using the f0 input allows for more freedom of detailed correction than the midi pitch input. The midi pitch we use is expressed as a frequency band divided into 88 constant proportions of the piano scale. On the other hand, the f0 representation can input the whole range of real numbers in the Hz unit, so a more detailed expression than the semitone unit is possible. Therefore, to control a subtle difference smaller than a semitone, an encoder that can directly reflect the f0 input is needed.

Second, using the midi pitch unit input helps learn the singer's f0 expression style. If you design a model that accepts only f0 contour as input, you have to create detailed contour representation every time you create a new song. However, if the neural network is trained in such a way that it generates a singing signal with an actual natural f0 contour from the pitch input quantized to the note level, the model is trained to generate a natural pitch expression from the note sequence. Therefore, we judged that the model that allows both types of inputs has an advantage over the model that allows only one input and based on this, we introduced a dual-path pitch encoder.

5.5.2 Considerations for use in the actual music production process

Our proposed SVS model was designed considering the convenience of the actual commercial music production process using it. In the case of making a song that includes a singing voice recording, the singer and the producer generally give and receive various feedback to record the soundtrack. First, the singer interprets the musical score in her/his style and sings it for the given musical score. Based on the first recorded results, the producer presents not only the pitch and pronunciation for various sections but also specific suggestions such as stress, breath sounds, and expressions to the singer. Based on this request for correction, the singer re-songs the singing section according to the producer's intention. Through the repetition of the process, a singing voice that reflects the producer's intention well is created.

We designed the network so that this feedback can be reflected in producing a sound source using the song synthesis system. Similar to the fact that the producer does not dictate all the detailed elements in the actual recording process, we hoped that the singer could interpret the given song independently and determine the various expressive elements by themselves. To this end, we introduced a style token module that can predict style elements from the text and pitch contents of the song in advance over time. We also wanted to enable a feedback process that could be modified by fine-grained control over detailed requirements from already created songs. Therefore, we constructed a framework that can be regenerated by modifying the f0 contour and style token sequence obtained again from the generated result.

For our proposed model to be applied to more general situations, the following improvements are needed. First, predicting the style token sequence from the singing voice should be possible. The model proposed by us is a system that only has a unidirectional path from symbolic midi data to a singing sound source. If it is possible to infer frame-level style features corresponding to a singing voice, it is possible to modify and reproduce it in a situation where only the sound source exists. However, since the current model can modify only the singing voice that was initially created in the state in which the sheet music is given, an extension to the interactive model is required. Second, it should be possible to propose more various style modifications. The current model can only control the elements modeled by itself from the data due to the limitations of self-supervised training. However, the feedback between singers and producers during the actual recording process includes a much wider range of expressions than this. Therefore, follow-up studies are needed to obtain labels for various expressions and improve the model in an explanatory way to enable a more intuitive modification process.

5.6 Conclusion

We proposed a local style token module and a dual-path pitch encoder to design an SVS system capable of modeling and controlling various musical expressions. We confirmed that the LST token predicted from contents can be controlled to modify expressions such as intensity and breathing and that f0 contour can be controlled through DPE to express various singing techniques related to pitch control. Listening evaluations showed that the proposed model can generate a high-quality singing voices by reflecting the users' intention.

Chapter 6

Conclusion

6.1 Thesis summary

In this thesis, we explored methods to construct a high-quality singing voice synthesis system that can effectively control various singing features. First, an effective single-singer singing voice synthesis system that can independently control pronunciation and pitch in a given dataset was constructed using the source-filter theory that mimics the principle of human vocal organs. We designed a network based on a conditional autoregressive neural network with two independent decoders and confirmed that the pronunciation accuracy is higher than that of the comparative model. Also, by introducing adversarial training, we were able to generate a spectrogram closer to the actual distribution and confirmed that the quality of the sound source could be significantly improved.

Next, we conducted a study to expand the single-singer model to the multi-singer model to reflect the singer's identity effectively. We designed a singer identity encoder that can compress and express a singer's identity from a singer's singing sound source and then condition it to two independent decoders to show that the singing style and timbre can be controlled independently. By introducing a cross-generation method that can synthesize songs by intersecting the singing style and timbre of two different singers, we proved that more creative music production is possible. Also, by interpolating the identity embeddings of two different singers, we proposed a methodology that can design a new speaker's identity that has never existed before.

In the third study, we propose the self-supervised expression modeling methodology that can effectively model more detailed expression information that is not labeled. We developed the existing multi-singer model into an expressive singing voice synthesis model. We confirmed that the expression elements such as intensity and breath sound inherent in a given input sequence could be captured through the self-supervised local style token module. Also, We show that these style tokens can be controlled by section in the inference stage. In addition, we prove that it is possible to modify the generated results by adopting a dual-path pitch encoder structure that can use both midi pitch and f0 contour as inputs to enable free correction of the detailed pitch contour.

Through a series of studies, we developed an expressive multi-singer singing voice synthesis network with a quality that can be used in the actual music production industry. The contribution of this paper is summarized as follows:

- By designing a singing voice synthesis system based on the source-filter model, the pitch and pronunciation, which are the basic elements of singing, are independently modeled to enable data efficient training.
- We proposed a method to generate singing voices closer to the actual

sound source distribution by applying adversarial training to the singing voice synthesis system and confirmed that higher-quality sound sources could be generated.

- We propose a multi-singer singing voice synthesis system that separates and defines the speaker's identity by the singing style and the timbre and can control them.
- We proposed a more expressive and controllable singing synthesis system by introducing a self-supervised local style token that can learn expressive information such as unlabeled breath sounds and tones directly from data.
- By introducing a dual-path pitch encoder that can use both sheet music and actual f0 input, we have confirmed that it is possible to freely modify the f0 contour for the created song sound source.

6.2 Limitations and future work

6.2.1 Improvements to a faster and robust system

Our proposed model is based on an auto-regressive neural network. Since the auto-regressive model predicts future information through past information, it has the advantage of generating natural results over time. However, since it is a sequential generation methodology that cannot predict the following information if the previous information is not predicted, it has a limitation in that the generation speed is slow. Since music creators generally work based on a processor that could process a bunch of data in parallel, AR models have a limitation in that they cannot fully utilize the advantages of such processors. In addition, the auto-regressive model has a limitation in that it is difficult to robustly generate a long singing sequence because errors may be accumulated while sequentially predicting each frame. Therefore, we will try to expand to a non-AR model that can be generated quickly and robustly without losing the advantages of the AR model.

However, in the non-AR model, since the acoustic feature of a specific frame is determined only from the input condition signals without being affected by the previous frames, it is essential to provide a better-aligned input condition signal. As mentioned in section 3.5.1, since the data we are using does not contain phoneme level alignment and contains annotation errors, it will fail if we directly train a non-AR model using data with a noisy timing label. Therefore, to expand to the non-AR model, We need to 1) think about a method to secure better-aligned data or 2) devise a method to model the target acoustic feature well even if the timing of the input condition signal is not accurate.

In the case of phoneme-audio alignment studies, there have been many studies on voice compared to singing. Therefore, we can design the study to adopt the phoneme-audio alignment model, which shows good performance in a speech to singing. In particular, it is possible to design an alignment model suitable for singing, considering the difference between voice and singing. Singing is more likely to have a longer phoneme than normal voice, so it is necessary to widen the receptive field of the model. In addition, since singing is likely to appear with various pitches changing for one phoneme, it may be helpful to have a more robust feature engineering in the pitch. Lastly, compared to a normal voice, the pronunciation of singing does not exactly match the lyrics or changes in many cases, so an objective design that can accommodate such variance can be considered.

From another perspective, we should consider how to predict acoustic features from noisy time-label data correctly. Since there is a limit to improving the annotation accuracy or making a better alignment model, it is also important to design a model that models the singing well, even in such a noisy label. In order to allow the case where the model generates acoustic features according to the correct order of the conditional input signals, even if the conditional input signal and the target acoustic feature are not perfectly aligned in time during the training process, DTW objective [86] can be used instead of frame-wise spectrogram distance loss. As another method, we can consider how to correct the incorrect timing of the input signal inside the model. Generally, sibilants start earlier than the note onset timing, and the coda pronunciation occupies a long part of the note. For the model to learn this prior by itself, it is also possible to train a module that corrects misalignment by introducing a module that predicts each phoneme's start position and length from a given note sequence by itself.

6.2.2 Explainable and intuitive controllability

In this paper, we defined various elements that made up singing and designed a singing voice synthesis system that can control them. Using the proposed model, music producers can control signal-level elements such as pitch, pronunciation, voice, breath, and stress. However, if we look at the process of recording a singing voice when an actual producer and singer meet, it is common to repeat recording and editing with more abstract requests. For example, it is possible to request corrections such as 'modify the chorus to make it a little sadder,' 'remove the overall strength and make it a lighter voice.' In order to better imitate the actual vocal music production process and fully reflect the producers' requirements, explainable and more intuitive controllability that can respond to these kinds of requests is needed. Therefore, we do not simply analyze the elements constituting a singing voice at the signal level but go to multi-model approaches that combine tags marked with adjectives or natural language to extend the existing model to a system that encompasses higher controllability.

Unfortunately, there is not much data on the pairing of voices, singing, and descriptions of it so far. Nevertheless, because the description of the voice expressed using adjectives is abstract, much data is needed to model this ambiguous relationship. So we have to think about how we can collect this kind of data.

The simplest way is to hire an annotator to work on choosing the most appropriate adjective expression for a given voice. It is possible to present pairs of adjectives opposite each other (ex-light: dark), ask which concept the given voice is closer to, and then collect labels for them. A more common method is collecting and processing data about people's ratings or reviews of sound sources. By collecting a large amount of information, such as comments on the video streaming platform or review magazines, the singer and the adjectives that express it can be collected and used as training data. Such data may be difficult to collect with accurate standards because various ways of expression and subjective standards are involved. However nevertheless, it can be used as valuable data that can model human evaluation and appreciation of art.

6.2.3 Extensions to common speech synthesis tools

As explained in Section 2.4, the singing voice is similar to speech in many respects. In addition, speech utterance such as emotional speech and realistic acting have characteristics that are closer to singing. If speech synthesis that can control the prosody is possible, it will be possible to expand it to more cultural and artistic contents beyond simply using it as a reading system. And if the framework proposed in singing voice synthesis is applied to general speaking voice, this extension is sufficiently possible. Therefore, based on the insights obtained in this study, we plan to conduct a study to expand it to a more general speech synthesis framework including both speech and singing voice.

A similar format is required to handle speech data together with singing data. In other words, a method is needed to express speech data not only through text, but as musical score. We propose a method of quantizing the f0 contour obtained from speech to midify speech. Through this, a more general speech synthesis system can be designed if the musical score representation for a given speech is obtained and the speech and singing are trained through one unified framework.

As a result of the initial experiments, we confirmed that the above method actually works properly, and showed that it is possible to generate general speech in the form of musical score. Interestingly, we observed that the speaker identity vector contained information about speaking type, speech and singing. According to the observation results, when creating a singing voice using the speaker identity extracted from the speech voice, the song was sang in a rap or reading style. Conversely, when a speech voice is synthesized using the speaker identity extracted from the singing voice, the voice is created as if it were a musical actor singing.

By analyzing these characteristics more sophisticatedly, we intend to develop the system as a tool that can create not only singing and speech, but also various genres of voice content such as poetry, musical, and rap that lie on the border between them.

6.2.4 Towards a collaborative and creative tool

Throughout this study, we focused on designing a system that faithfully reflects a given user's input and generates a singing voice. However, to be more creative in making works of art, models need to go beyond just producing accurate results quickly. Therefore, we propose some tasks to use the proposed model as a more collaborative tool in the singing voice synthesis task and try to solve it through follow-up studies:

Novel voice design We paid attention to the voice design task that can freely design the singer's identity, one of the essential elements of composing a song. Based on the analysis of various singers' timbre and singing styles, more creative work is possible if there is a technology that can directly create them. For this, we can consider a method of manipulating the singer's identity embedding. We can use a method such as [87] to manipulate the singer identity embedding in the latent space. For more intuitive control, a method for labeling attributes such as the singer's age, gender, confident pitch, genre, and style and training a neural network linking the attribute space and the singer identity embedding space can also be considered. In this case, it may be possible to sample a new identity that satisfies the desired attribute or manipulate an attribute of an already given identity.

Emotion from lyrics We can think of a lyric to emotion task that generates a musical expression appropriate for the content from the lyric information. Many singers sing songs based on their understanding of the mood and lyrics of the song. Combining this with language modeling to understand the relationship between musical expression and lyrics will help more emotional singing synthesis. More specifically, it is possible to consider a method of training an SVS model with the emotion label after acquiring an emotion label according to a given singing section by using a language model that infers emotion from lyrics. If it is possible to directly control elements such as changes in energy according to emotions and expression of pitch according to emotions, it will be possible to create songs with more realistic emotional expressions.

Unconditional SVS Throughout this study, we argued that musical score was necessary to determine singing. However, people can sing without a musical score. Therefore, an unconditional singing synthesis task can be considered. Its medium can be the emotion of a scene or a moment, and humans can sing while humming through their unique creative process. If the scope of research is expanded to the field of singing synthesis in a state where the given condition is not perfect, it is thought that various results that can inspire creators can be explored. For this, we can try the method of masked language modeling. If the SVS model is trained using a partially deleted score sequence, it is possible to train a model that understands the context of a given score and generates an appropriate singing segment for the masked part. Furthermore, it is possible to consider a method of training the SVS model by gradually abstracting the level of the score, such as the chord sequence instead of the score and the global key of the song instead of the chord sequence. This will be possible to secure a model that creatively generates parts other than condition information by gradually narrowing the restricted area through conditional input.

Bibliography

[1] H. Tachibana, K. Uenoyama, and S. Aihara, “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)*. IEEE, 2018, pp. 4784–4788.

[2] Z. Duan, H. Fang, B. Li, K. C. Sim, and Y. Wang, “The nus sung and spoken lyrics corpus: A quantitative comparison of singing and speech,” in *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*. IEEE, 2013, pp. 1–9.

[3] S. Choi, W. Kim, S. Park, S. Yong, and J. Nam, “Children’s song dataset for singing voice research,” in *International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[4] H. Tamaru, S. Takamichi, N. Tanji, and H. Saruwatari, “Jvs-music: Japanese multispeaker singing-voice corpus,” *arXiv preprint arXiv:2001.07044*, 2020.

[5] I. Ogawa and M. Morise, “Tohoku kiritan singing database: A singing database for statistical parametric singing synthesis using japanese pop

songs," *Acoustical Science and Technology*, vol. 42, no. 3, pp. 140–145, 2021.

[6] Y. Wang, X. Wang, P. Zhu, J. Wu, H. Li, H. Xue, Y. Zhang, L. Xie, and M. Bi, "Opencpop: A high-quality open source chinese popular song corpus for singing voice synthesis," *arXiv preprint arXiv:2201.07429*, 2022.

[7] 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," *arXiv preprint arXiv:1904.02882*, 2019.

[8] J. Yamagishi, C. Veaux, K. MacDonald *et al.*, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92)," 2019.

[9] E. Bakhturina, V. Lavrukhin, B. Ginsburg, and Y. Zhang, "Hi-fi multi-speaker english tts dataset," *arXiv preprint arXiv:2104.01497*, 2021.

[10] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *The journal of the acoustical society of america*, vol. 8, no. 3, pp. 185–190, 1937.

[11] T. Chiba and M. Kajiyama, *The vowel: Its nature and structure.* Phonetic society of Japan Tokyo, 1958, vol. 652.

[12] A. v. d. 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.

[13] Y. Wang, R. 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.

[14] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, “Natural tts synthesis by conditioning wavenet on mel spectrogram predictions,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2018, pp. 4779–4783.

[15] R. 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,” *arXiv preprint arXiv:1803.09047*, 2018.

[16] Y. Wang, D. Stanton, Y. Zhang, R. 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,” *arXiv preprint arXiv:1803.09017*, 2018.

[17] W.-N. Hsu, Y. Zhang, R. J. Weiss, Y.-A. Chung, Y. Wang, Y. Wu, and J. Glass, “Disentangling correlated speaker and noise for speech synthesis via data augmentation and adversarial factorization,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2019, pp. 5901–5905.

[18] L. Zhang, C. Yu, H. Lu, C. Weng, Y. Wu, X. Xie, Z. Li, and D. Yu, “Learning singing from speech,” *arXiv preprint arXiv:1912.10128*, 2019.

[19] F. Biadsy, R. J. Weiss, P. J. Moreno, D. Kanevsky, and Y. Jia, “Parrotron: An end-to-end speech-to-speech conversion model and its applica-

tions to hearing-impaired speech and speech separation," *arXiv preprint arXiv:1904.04169*, 2019.

[20] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

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

[22] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi." in *Interspeech*, vol. 2017, 2017, pp. 498–502.

[23] J. Kim, S. Kim, J. Kong, and S. Yoon, "Glow-tts: A generative flow for text-to-speech via monotonic alignment search," *Advances in Neural Information Processing Systems*, vol. 33, pp. 8067–8077, 2020.

[24] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *International Conference on Machine Learning*. PMLR, 2021, pp. 5530–5540.

[25] A. Gibiansky, S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep voice 2: Multi-speaker neural text-to-speech," in *Advances in neural information processing systems*, 2017, pp. 2962–2970.

[26] Y. Chen, Y. Assael, B. Shillingford, D. Budden, S. Reed, H. Zen, Q. Wang, L. C. Cobo, A. Trask, B. Laurie *et al.*, "Sample efficient adaptive text-to-speech," *arXiv preprint arXiv:1809.10460*, 2018.

[27] M. Chen, X. Tan, B. Li, Y. Liu, T. Qin, S. Zhao, and T.-Y. Liu, "Adaspeech: Adaptive text to speech for custom voice," *arXiv preprint arXiv:2103.00993*, 2021.

[28] G. Pamisetty and K. Murty, "Prosody-tts: An end-to-end speech synthesis system with prosody control," *arXiv preprint arXiv:2110.02854*, 2021.

[29] C.-M. Chien and H.-y. Lee, "Hierarchical prosody modeling for non-autoregressive speech synthesis," in *2021 IEEE Spoken Language Technology Workshop (SLT).* IEEE, 2021, pp. 446–453.

[30] M. Blaauw and J. Bonada, "A neural parametric singing synthesizer modeling timbre and expression from natural songs," *Applied Sciences*, vol. 7, no. 12, p. 1313, 2017.

[31] K. Nakamura, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Singing voice synthesis based on convolutional neural networks," *arXiv preprint arXiv:1904.06868*, 2019.

[32] K. Nakamura, S. Takaki, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Fast and high-quality singing voice synthesis system based on convolutional neural networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2020, pp. 7239–7243.

[33] J. Kim, H. Choi, J. Park, S. Kim, J. Kim, and M. Hahn, "Korean singing voice synthesis system based on an lstm recurrent neural network," in *INTERSPEECH 2018.* International Speech Communication Association, 2018.

[34] Y.-H. Yi, Y. Ai, Z.-H. Ling, and L.-R. Dai, "Singing voice synthesis using deep autoregressive neural networks for acoustic modeling," *arXiv preprint arXiv:1906.08977*, 2019.

[35] P. Chandna, M. Blaauw, J. Bonada, and E. Gomez, "Wgansing: A multi-voice singing voice synthesizer based on the wasserstein-gan," *arXiv preprint arXiv:1903.10729*, 2019.

[36] F. Chen, R. Huang, C. Cui, Y. Ren, J. Liu, and Z. Zhao, "Singgan: Generative adversarial network for high-fidelity singing voice generation," *arXiv preprint arXiv:2110.07468*, 2021.

[37] J. Chen, X. Tan, J. Luan, T. Qin, and T.-Y. Liu, "Hifisinger: Towards high-fidelity neural singing voice synthesis," *arXiv preprint arXiv:2009.01776*, 2020.

[38] Y. Zhang, J. Cong, H. Xue, L. Xie, P. Zhu, and M. Bi, "Visinger: Variational inference with adversarial learning for end-to-end singing voice synthesis," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2022, pp. 7237–7241.

[39] J. Liu, C. Li, Y. Ren, F. Chen, P. Liu, and Z. Zhao, "Diffsinger: Diffusion acoustic model for singing voice synthesis," *arXiv preprint arXiv:2105.02446*, 2021.

[40] Y. Ren, X. Tan, T. Qin, J. Luan, Z. Zhao, and T.-Y. Liu, "Deepsinger: Singing voice synthesis with data mined from the web," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1979–1989.

[41] R. Valle, J. Li, R. Prenger, and B. Catanzaro, “Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2020, pp. 6189–6193.

[42] H. Xue, S. Yang, Y. Lei, L. Xie, and X. Li, “Learn2sing: Target speaker singing voice synthesis by learning from a singing teacher,” in *2021 IEEE Spoken Language Technology Workshop (SLT).* IEEE, 2021, pp. 522–529.

[43] M. Blaauw, J. Bonada, and R. Daido, “Data efficient voice cloning for neural singing synthesis,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2019, pp. 6840–6844.

[44] G.-H. Lee, T.-W. Kim, H. Bae, M.-J. Lee, Y.-I. Kim, and H.-Y. Cho, “N-singer: A non-autoregressive korean singing voice synthesis system for pronunciation enhancement,” *arXiv preprint arXiv:2106.15205*, 2021.

[45] J. Tae, H. Kim, and Y. Lee, “Mlp singer: Towards rapid parallel korean singing voice synthesis,” in *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP).* IEEE, 2021, pp. 1–6.

[46] J.-Y. Liu, Y.-H. Chen, Y.-C. Yeh, and Y.-H. Yang, “Score and lyrics-free singing voice generation,” *arXiv preprint arXiv:1912.11747*, 2019.

[47] K. Markopoulos, N. Ellinas, A. Vioni, M. Christidou, P. Kakoulidis, G. Vamvoukakis, G. Maniati, J. S. Sung, H. Park, P. Tsiakoulis *et al.*,

"Rapping-singing voice synthesis based on phoneme-level prosody control," *arXiv preprint arXiv:2111.09146*, 2021.

[48] M. Macon, L. Jensen-Link, E. B. George, J. Oliverio, and M. Clements, "Concatenation-based midi-to-singing voice synthesis," in *Audio Engineering Society Convention 103.* Audio Engineering Society, 1997.

[49] J. Bonada, A. Loscos, O. Mayor, and H. Kenmochi, "Sample-based singing voice synthesizer using spectral models and source-filter decomposition," in *Third International Workshop on Models and Analysis of Vocal Emissions for Biomedical Applications*, 2003.

[50] H. Kenmochi and H. Ohshita, "Vocaloid-commercial singing synthesizer based on sample concatenation," in *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[51] M. Nishimura, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Singing voice synthesis based on deep neural networks." in *Interspeech*, 2016, pp. 2478–2482.

[52] Y.-A. Chung, Y. Wang, W.-N. Hsu, Y. Zhang, and R. Skerry-Ryan, "Semi-supervised training for improving data efficiency in end-to-end speech synthesis," *arXiv preprint arXiv:1808.10128*, 2018.

[53] S. Li, S. Villette, P. Ramadas, and D. J. Sinder, "Speech bandwidth extension using generative adversarial networks," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2018, pp. 5029–5033.

[54] T. Miyato and M. Koyama, “cgans with projection discriminator,” *arXiv preprint arXiv:1802.05637*, 2018.

[55] A. Brock, J. Donahue, and K. Simonyan, “Large scale gan training for high fidelity natural image synthesis,” *arXiv preprint arXiv:1809.11096*, 2018.

[56] 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.

[57] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[58] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of wasserstein gans,” in *Advances in Neural Information Processing Systems*, 2017, pp. 5767–5777.

[59] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” *arXiv preprint arXiv:1802.05957*, 2018.

[60] L. Mescheder, A. Geiger, and S. Nowozin, “Which training methods for gans do actually converge?” *arXiv preprint arXiv:1801.04406*, 2018.

[61] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[62] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[63] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[64] 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.

[65] J. Sundberg, "Level and center frequency of the singer's formant," *Journal of voice*, vol. 15, no. 2, pp. 176–186, 2001.

[66] T. F. Cleveland, "Acoustic properties of voice timbre types and their influence on voice classification," *The Journal of the Acoustical Society of America*, vol. 61, no. 6, pp. 1622–1629, 1977.

[67] J. Lee, H.-S. Choi, C.-B. Jeon, J. Koo, and K. Lee, "Adversarially trained end-to-end korean singing voice synthesis system," *Proc. Interspeech 2019*, pp. 2588–2592, 2019.

[68] K. Saino, H. Zen, Y. Nankaku, A. Lee, and K. Tokuda, "An hmm-based singing voice synthesis system," in *Ninth International Conference on Spoken Language Processing*, 2006.

[69] K. Nakamura, K. Oura, Y. Nankaku, and K. Tokuda, "Hmm-based singing voice synthesis and its application to japanese and english," in *2014*

IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2014, pp. 265–269.

[70] G. D. Ruxton, "The unequal variance t-test is an underused alternative to student's t-test and the mann–whitney u test," *Behavioral Ecology*, vol. 17, no. 4, pp. 688–690, 2006.

[71] A. Noguchi and T. Harada, "Image generation from small datasets via batch statistics adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2750–2758.

[72] S. Zhang, C.-T. Do, R. Doddipatla, and S. Renals, "Learning noise invariant features through transfer learning for robust end-to-end speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2020, pp. 7024–7028.

[73] S. Mo, M. Cho, and J. Shin, "Freeze the discriminator: a simple baseline for fine-tuning gans," *arXiv preprint arXiv:2002.10964*, 2020.

[74] X. Zhuang, T. Jiang, S.-Y. Chou, B. Wu, P. Hu, and S. Lui, "Litesing: Towards fast, lightweight and expressive singing voice synthesis," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2021, pp. 7078–7082.

[75] Y. Hono, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Sinsy: A deep neural network-based singing voice synthesis system," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2021.

[76] C.-Z. A. Huang, H. V. Koops, E. Newton-Rex, M. Dinculescu, and C. J. Cai, “Ai song contest: Human-ai co-creation in songwriting,” *arXiv preprint arXiv:2010.05388*, 2020.

[77] Y. Wada, R. Nishikimi, E. Nakamura, K. Itoyama, and K. Yoshii, “Sequential generation of singing f0 contours from musical note sequences based on wavenet,” in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC).* IEEE, 2018, pp. 983–989.

[78] Y. Ohishi, H. Kameoka, D. Mochihashi, and K. Kashino, “A stochastic model of singing voice f0 contours for characterizing expressive dynamic components,” in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[79] S. W. Lee, M. Dong, and H. Li, “A study of f0 modelling and generation with lyrics and shape characterization for singing voice synthesis,” in *2012 8th International Symposium on Chinese Spoken Language Processing.* IEEE, 2012, pp. 150–154.

[80] J. Bonada and M. Blaauw, “Hybrid neural-parametric f0 model for singing synthesis,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2020, pp. 7244–7248.

[81] J. Lee, H.-S. Choi, J. Koo, and K. Lee, “Disentangling timbre and singing style with multi-singer singing synthesis system,” in *ICASSP 2020-2020*

IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020, pp. 7224–7228.

[82] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International conference on machine learning.* PMLR, 2017, pp. 933–941.

[83] H. Kameoka, K. Yoshizato, T. Ishihara, Y. Ohishi, K. Kashino, and S. Sagayama, "Generative modeling of speech f0 contours." in *INTERSPEECH.* Citeseer, 2013, pp. 1826–1830.

[84] Y. Ikemiya, K. Itoyama, and H. G. Okuno, "Transferring vocal expression of f0 contour using singing voice synthesizer," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems.* Springer, 2014, pp. 250–259.

[85] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *arXiv preprint arXiv:2010.05646*, 2020.

[86] M. Cuturi and M. Blondel, "Soft-dtw: a differentiable loss function for time-series," in *International conference on machine learning.* PMLR, 2017, pp. 894–903.

[87] Y. Shen and B. Zhou, "Closed-form factorization of latent semantics in gans," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1532–1540.

초 록

가창 합성은 주어진 입력 악보로부터 자연스러운 가창 음성을 합성해내는 것을 목표로 한다. 가창 합성 시스템은 음악 제작 비용을 크게 줄일 수 있을 뿐만 아니라 창작자의 의도를 보다 쉽고 편리하게 반영할 수 있도록 돕는다. 하지만 이러한 시스템의 설계를 위해서는 다음 세 가지의 도전적인 요구사항이 존재한다. 1) 가창을 이루는 다양한 요소를 독립적으로 제어할 수 있어야 한다. 2) 높은 품질 수준 및 사용성을 달성해야 한다. 3) 충분한 훈련 데이터를 확보하기 어렵다. 이러한 문제에 대응하기 위해 우리는 대표적인 음성 생성 모델링 기법인 소스-필터 이론에 주목하였다. 가창 신호를 음정 정보에 해당하는 소스와 발음 정보에 해당하는 필터의 합성곱으로 정의하고, 이를 각각 독립적으로 모델링할 수 있는 구조를 설계하여 훈련 데이터 효율성과 제어 가능성을 동시에 확보하고자 하였다. 또한 우리는 발음, 음정, 화자 등 조건부 입력이 주어진 상황에서 시계열 데이터를 효과적으로 모델링하기 위하여 조건부 자기회귀 모델 기반의 심층신경망을 활용하였다. 마지막으로 레이블링 되어있지 않은 음악적 표현을 모델링할 수 있도록 우리는 자기지도학습 기반의 스타일 모델링 기법을 제안했다. 우리는 제안한 모델이 발음, 음정, 음색, 창법, 표현 등 다양한 요소를 유연하게 제어하면서도 실제 가창과 구분이 어려운 수준의 고품질 가창 합성이 가능함을 확인했다. 나아가 실제 음악 제작 과정을 고려한 생성 및 수정 프레임워크를 제안하였고, 새로운 목소리 디자인, 교차 생성 등 창작자의 상상력과 한계를 넓힐 수 있는 응용이 가능함을 확인했다.

주요어: 가창 합성, 소스-필터 이론, 조건부 자기회귀형 인공 신경망, 가수의 특징, 음악적 표현

학 번: 2018-23798

감사의 글

2018년 봄, 음악오디오 연구실에 입학한 뒤 저에게 일어난 수 많은 경험들은 그 어느때보다 많이 저를 성장시켰습니다. 다양한 주제의 연구과제를 수행하고, 서로 다른 배경을 가진 학우들과 논의하며, 흥미로운 연구를 원없이 진행할 수 있었던 것은 제 삶 속에서 가장 값진 시간이 이었습니다. 아무것도 모르던 새내기 대학원생을 진심을 담아 지도해주시어 혼자서도 연구를 수행할 힘을 길러주신 이교구 교수님께 감사드립니다. 부족함이 많던 동료였음에도 항상 아낌없는 피드백과 조언 주시고 함께해 주셨던 MARG 선후배분들께 감사합니다. 오랜 시간 학문의 길을 걸어가는 동안 언제나 항상 든든하게 제 길을 존중하고 응원해주셨던 가족 및 친지 여러분들에게 감사합니다. 모두의 도움이 없었더라면 이러한 결실에 도달하기 어려웠을 것이라 생각합니다. 다시 한 번 모두에게 감사의 말씀을 드립니다.

그동안 받았던 수 많은 도움에 보답할 수 있도록, 저는 제가 배운 것들을 통해 사람들을 행복하게 하는 기술을 만드는 연구자의 길을 진심을 다해 걸어나가고자 합니다. 많은 사람들의 행복한 삶을 위해 해결되어야 하는 문제들에 대해 끊임없이 고민하고 그것들을 해결함으로서 저 또한 행복해질 수 있는 삶을 살기 위해 노력하겠습니다. 음악을 만드는 모든 창작자들을 존중하고 그들을 지원할 수 있는 기술을 연구하겠습니다. 음악을 사랑하는 모든 청취자들이 더욱 풍부하고 다양한 컨텐츠를 향유할 수 있도록 노력하겠습니다. 끝이 아닌 새로운 시작이라는 생각으로, 힘차게 나아가겠습니다. 모두에게 다시 한 번 감사합니다.

2022년 8월,

이 주 헌