FT-GAN: Fine-Grained Tune Modeling for Chinese Opera Synthesis

Meizhen Zheng[1,2*], **Peng Bai**[1,2*], **Xiaodong Shi**[1,2†], **Xun Zhou**[1], **Yiting Yan**[1]

[1]Department of Artificial Intelligence, School of Informatics, Xiamen University, Xiamen, China
[2]Key Laboratory of Digital Protection and Intelligent Processing of Intangible Cultural Heritage of Fujian and Taiwan (Xiamen University), Ministry of Culture and Tourism, China
{midon, baipeng}@stu.xmu.edu.cn, mandel@xmu.edu.cn, {xzhou, eatingyan}@stu.xmu.edu.cn

Abstract

Although singing voice synthesis (SVS) has made significant progress recently, with its unique styles and various genres, Chinese opera synthesis requires greater attention but is rarely studied for lack of training data and high expressiveness. In this work, we build a high-quality Gezi Opera (a type of Chinese opera popular in Fujian and Taiwan) audio-text alignment dataset and formulate specific data annotation methods applicable to Chinese operas. We propose FT-GAN, an acoustic model for fine-grained tune modeling in Chinese opera synthesis based on the empirical analysis of the differences between Chinese operas and pop songs. To further improve the quality of the synthesized opera, we propose a speech pre-training strategy for additional knowledge injection. The experimental results show that FT-GAN outperforms the strong baselines in SVS on the Gezi Opera synthesis task. Extensive experiments further verify that FT-GAN performs well on synthesis tasks of other operas such as Peking Opera. Audio samples, the dataset, and the codes are available at https://zhengmidon.github.io/FTGAN.github.io/.

Introduction

Opera is a unique genre of singing with regional characteristics. There are many famous operas worldwide, such as Italian operas and Chinese operas. Chinese opera is a stage performing art that utilizes various performance techniques (including singing, reciting, acting, and acrobatic fighting). With its long history, it has developed a unique system of role types (the male lead, the female lead, the painted face, and the clown) and a wide variety of tunes. In addition, Chinese operas have been influenced by different regional cultures to evolve into more diverse types. The most direct manifestation of the regional nature of opera is the use of dialects for singing, for example, Gezi Opera is sung in Hokkien. These characteristics make opera synthesis significant and yet challenging at the same time.

Just as many datasets of pop song (Wang et al. 2022b; Tamaru et al. 2020; Zhang et al. 2022a; Wilkins et al. 2018) have been released one after another, the models and methods of singing voice synthesis have been continuously developed (Chen et al. 2020; Lee et al. 2019; Kim, Kang, and Lee 2022; Wu and Shi 2023), and the synthesized singing voice has been natural enough and relatively expressive. However, despite the profound cultural heritage and broad audience base of Chinese operas, the research on automatic opera synthesis has lagged behind compared to singing voice synthesis and speech synthesis due to the scarcity of publicly available high-quality opera datasets. Therefore, to fill the gap to some extent and to promote the research on opera synthesis, in this paper, we take the most famous opera in Fujian and Taiwan as the primary research focus and construct an accurately annotated Gezi Opera Audio-Text alignment dataset (GOAT), which can be used for opera synthesis research with controllability in lyrics, role, tune, and other potential aspects.

Chinese operas differ from pop songs in that their rhythmic changes are more complex and diverse than pop songs (Wu et al. 2020). Specifically, some characters in a Chinese opera aria often undergo multiple pitch changes (also known as slurs), while pop songs usually follow a single pitch for each character. In addition, the duration variance of phonemes in Chinese operas is so large that sometimes even a single phoneme can be sung for tens of seconds. These fine-grained tune styles incur difficulties in opera synthesis. In this work, we attempt to overcome these difficulties through data annotation and model architecture design. Specifically, for the voice characteristics of Gezi Opera (which also exist in other Chinese operas), we design two data annotation strategies, namely, vowel morphing and vocal run, to align and annotate the data more precisely. In terms of modeling, although some singing voice synthesis models such as FastSpeech2 (Ren et al. 2020) and DiffSinger (Liu et al. 2022) perform well in singing voice synthesis tasks, their synthesized operas are unsatisfactory because they do not model the fine-grained tunes typical of opera. In this work, we propose FT-GAN, an acoustic model designed for fine-grained tune modeling in Chinese opera synthesis. In addition, from the perspective of pronunciation and linguistics, we abstract the general Hokkien speech into specific Gezi Opera and propose a speech pre-training strategy further to improve the performance and robustness of the model. The experimental results show that FT-GAN outperforms the strong baselines in SVS on the Gezi Opera synthesis task. Extensive experiments further verify that FT-GAN has good performance on the Peking Opera synthesis task as

*These authors contributed equally.
†Corresponding author.

well.

The main contributions of this work are summarized as follows:

- We build GOAT, the first precisely annotated high-quality Gezi Opera audio-text alignment dataset that can be used for opera synthesis research with controllability in lyrics, role, tune, and other potential aspects. Furthermore, we propose two annotation strategies specialized for Chinese opera dataset construction.
- We propose an acoustic model, FT-GAN, and a speech pre-training strategy that helps improve the model's performance and robustness. FT-GAN is better at modeling fine-grained tunes in Chinese operas than the baseline SVS models.
- Extensive experiments on the annotation strategies, the model architectures, and Peking Opera synthesis verify the generalization and effectiveness of our methods.

Related Works

Singing Voice Synthesis

In the early stage, traditional concatenative methods (Macon et al. 1997; Kenmochi and Ohshita 2007) and statistical parametric methods (Oura et al. 2010; Saino et al. 2006) are used in singing voice synthesis. Although these two families of methods can generate good songs, they suffer from a lack of flexibility and scalability. With the development of deep learning, more and more methods try to synthesize singing voices using deep neural networks. Mainstream SVS models based on deep neural networks can be categorized into two categories: autoregressive and non-autoregressive. The autoregressive models (Gu et al. 2021; Yang et al. 2021; Lee et al. 2019; Hono et al. 2023; Nishihara et al. 2023; Yi et al. 2019; Wang et al. 2022a) are conditioned on previously generated information at each step of generation, and thus does not require aligned text-speech pair data for training. However, they have problems such as exposure bias and slow inference. Notably, most autoregressive SVS models at this stage are based on the Tacotron (Wang et al. 2017). The other methods, especially those based on Feed-Forward Transformers (FFTs) (Vaswani et al. 2017), are non-autoregressive. Non-autoregressive models have the advantage that the generation process is easy to control and allows for fast inference but generally require accurately annotated text-speech data for supervised training, which incurs significant costs. Up to now, various generative models have been used to generate singing voices in a non-autoregressive way. Generative adversarial networks (GAN) (Goodfellow et al. 2020) can effectively alleviate the over-smoothing problem caused by L1 or L2 losses and generate high-fidelity audio (Lee et al. 2019; Wu and Luan 2020; Lee et al. 2020; Chen et al. 2020; Zhuang et al. 2021; Cho et al. 2022; Kim et al. 2022; Lee et al. 2021; Zhang et al. 2022d; Lu et al. 2020). VITS (Kim et al. 2021) is a adversarial trained Variational Autoencoder (VAE) (Kingma and Welling 2013) model, which has achieved remarkable results in speech synthesis tasks. Some works apply it to SVS tasks and achieve good performances (Lei et al. 2022; Zhang et al. 2022b,c). Inspired by the excellent performance of diffusion models in image generation tasks, more and more methods for high-quality singing voice synthesis using diffusion models have been proposed (Liu et al. 2022; Ye et al. 2023; Shen et al. 2023; Wu and Shi 2023).

Peking Opera Synthesis

Although models of singing voice synthesis have been increasingly well-developed, there has been limited research on opera synthesis. Gong et al. (2017) collect a dataset of 120 Peking Opera arias sung by professional actors and amateurs. The total audio duration is 7 hours, but only 1.7 hours of them have been aligned and annotated. Wu et al. (2019) use this dataset to train a Peking Opera synthesis model based on the DurIAN (Yu et al. 2019) architecture with musical scores and lyrics as inputs. Subsequently, to tackle the problem of significant deviation between the pitch of the actual singing of opera and the musical score, a melody transcription method is proposed to create a pseudo musical score in place of the pitch of the original musical score as an input to the model (Wu et al. 2020). In addition, a Lagrange-multiplier-constrained mixture density network (MDN) is applied to the duration predictor to improve the accuracy of phoneme duration prediction. However, there is still much room for improvement in this model regarding the quality, naturalness, and expressiveness of synthesized audio.

Dataset Construction

Data Collection

We invite five professional Gezi Opera actors to our data recording sessions. These actors (including three female actors and two male actors) have a high professional level and rich stage performance experience. Each actor is required to perform their familiar arias. Due to various constraints, we cannot arrange for actors to use professional recording equipment to record audio. Therefore, the actors do all the recording work using their mobile phones and headphones. Even so, we find that the recorded audio quality can still meet the requirements for training the Gezi Opera acoustic model because: 1) We request that the sampling rate for recording audio should be set to 48kHz. 2) The recording sound quality of modern mobile phones and headphones is high enough to ensure an excellent auditory experience. Finally, we obtain 1938 audio clips with durations ranging from 1.36 seconds to 29.89 seconds. The total audio duration is 4.54 hours.

Table 1 shows more detailed information for each actor. Each actor has an intact and similar pitch range which means we can train a model with the data that can sing enough notes for each voice. The data on the role type *clown* is relatively limited, but it is also enough to train his voice.

Annotations

Role Type and Tune Annotation To promote more comprehensive research on Chinese opera synthesis in lyrics, role, tune, and other potential aspects, we annotate the role type and tune information on the file name of each audio piece.

Gender	Singer ID	Major Role Type	Pitch Range	Hours
Female	dan-1	female lead	47-78 (B2, 123.5Hz - F#5, 739.9Hz)	0.98
	dan-2	female lead	43-78 (G2, 97.9Hz - F#5, 739.9Hz)	0.98
	dan-3	female lead	44-77 (G#2, 103.8Hz - F5, 698.5Hz)	0.99
Male	chou	clown	43-74 (G2, 97.9Hz - D5, 587.3Hz)	0.57
	sheng	male lead	43-74 (G2, 97.9Hz - D5, 587.3Hz)	1.02

Table 1: The information of singers.

Phoneme Annotation To annotate the text to the phoneme level for model training, the recognized *Taiwan Roman Phoneme Scheme*[1] is used as the phoneme annotation scheme in our work. We first use an online tool[2] to convert the recorded Chinese lyrics into phonemes. Then, the following steps of annotation are taken:

1. The annotators are required to check the correctness of each phoneme in the automatically converted phoneme sequence.
2. A special phoneme *sp* is added to the phoneme set, representing the blank parts in the audio (including pauses and breaths). The annotators are required to add *sp* to the phoneme sequence at every blank position.
3. In opera audio, sometimes a vowel may last for a relatively long time (e.g., several seconds). In this condition, the vowel may not just be pronounced as itself, some similar but different vowels may be pronounced before or after it. For example, the vowel *ai* may be pronounced as the following sequence: *a ai i*. We notice this phenomenon and propose the **vowel morphing** annotation strategy. This strategy requires the annotators to pay attention to the vowels sung for a long time and add extra vowels to the original phoneme sequence according to the true pronunciation.
4. Chinese operas generally have more complex rhythmic changes than pop songs, which is reflected in the situation that a single character can be pronounced in multiple pitches. The annotators are required to replicate a phoneme in the phoneme sequence if it keeps pronunciation unchanged but has significant changes in pitch. Here we take an annotation for example: *a a a / A3 B3 D4*. The first part separated by the slash, *a*, is the phoneme annotation, and the second part represents its pitch change. The phoneme *a* is replicated twice because it has two pitch changes. We call this strategy **vocal run**.

These annotation steps lead to the phoneme sequence $\mathcal{T}$.

Alignment Annotation To save time and labor costs, while maximizing the accuracy of alignment annotation, we combine automated preliminary annotation and manual checking in the alignment annotation. First, we use the Montreal Forced Aligner (MFA) (McAuliffe et al. 2017) to align the phoneme sequences with the audio pieces preliminarily. Then the annotators use Praat[3] to conduct careful manual checking. At this point, we obtain the duration sequence $\mathcal{D}$ corresponding to the phoneme sequence.

Pitch Annotation As mentioned by Wu et al. (2020), improvisation and personal style variation are widely present in Chinese opera performances which results in significant differences between the actual singing pitches and score pitches. We need to annotate the true pitches of the audio, rather than the pitches in the musical score. According to the usual practice, we can invite professional music practitioners to assist us in pitch annotation, but manual annotation is time-consuming and unnecessary as there are already some precise and efficient automatic pitch extraction algorithms and tools (such as Parselmouth[4]) available for us to use. It should be noted that what we need to annotate here is not the complex and ever-changing real audio pitch, but a representative pitch within each phoneme pronunciation interval. To be precise, what this task requires to do is, given a known phoneme sequence $\mathcal{T} = \{t_i\}_{i=1}^m$ and a duration sequence $\mathcal{D} = \{d_i\}_{i=1}^m$, find the corresponding pitch sequence $\mathcal{P} = \{p_i\}_{i=1}^m$. Intuitively, the simplest way to find p_i is to take the average pitch value on duration d_i, i.e., $p_i = \frac{1}{s}\sum_{k=1}^{s} \dot{p}_k$, here we assume that in duration d_i the set of pitch sampling points is $\{\dot{p}_k\}_{k=1}^s$, whose size is s. This can work well for pitch annotation in general singing music such as pop songs, as the pitch curve of general singing music is relatively flat and the pitch does not change frequently. However, for Chinese operas, this simple method can result in certain errors as there are many irregular changes and steep slopes in the pitch curve. To solve this problem, we propose Algorithm 1. The key idea of this algorithm is to use statistical data to identify and eliminate outliers, and then calculate the mean on the clean set. After this annotation, we obtain the pitch sequence $\mathcal{P}$.

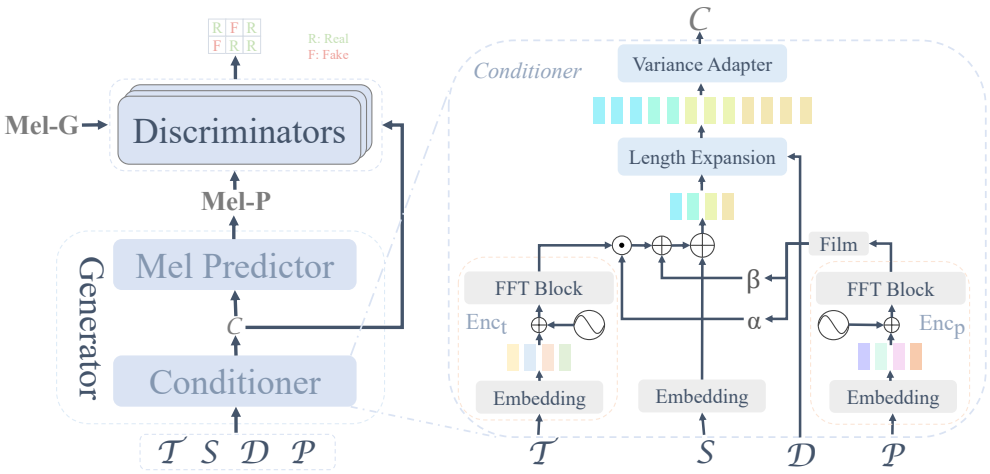
Audio Content Coverage Gezi Opera mainly includes four role types, namely, the male lead, the female lead, the painted face, and the clown, in which the male lead and the female lead are the two main role types. Different role types represent different singing styles. The recorded audio pieces contain the following three role types: the male lead, the female lead, and the clown. The painted face is not included, but it does not matter because it is rarely seen in Gezi Opera. The tune is the most representative and distinctive element of Gezi Opera which includes three major types, the seven-

[1]https://language.moe.gov.tw/result.aspx?classify_sn=42&subclassify_sn=446&thirdclassify_sn=486&content_sn=8

[2]http://tts001.iptcloud.net:8804/

[3]https://www.fon.hum.uva.nl/praat/

[4]https://github.com/YannickJadoul/Parselmouth



(a) Our model **(b) The Conditioner**

Figure 1: (a) The overall architecture of the model, where Mel-P indicates the predicted mel-spectrogram and Mel-G indicates the ground-truth mel-spectrogram. $\mathcal{T}$, $\mathcal{S}$, $\mathcal{D}$, and $\mathcal{P}$ are the phoneme sequence, the speaker label, the duration sequence, and the pitch sequence. $\mathcal{C}$ is the unified conditional representation of these four conditional inputs. (b) The architecture of the conditioner, where Enc_t indicates the phoneme encoder, and Enc_p indicates the pitch encoder. The film is an information-integrating module.

Algorithm 1: Pseudo Pitch Extraction Algorithm

Input: Pitch sampling point sequence $\mathbf{P} = \{\dot{p}_k\}_{k=1}^{s}$
Parameter: Retention ratio coefficient ϵ, sample point threshold $\mathbb{T}$
Output: Pitch value p

$\mu = \frac{1}{s}\sum_{k=1}^{s}\dot{p}_k, \sigma = \sqrt{\frac{1}{s}\sum_{k=1}^{s}(\dot{p}_k - \mu)^2}$
Compute $\hat{\mathbf{P}} = \{\hat{p}_k : \hat{p}_k \in (\mu - \epsilon\sigma, \mu + \epsilon\sigma), \hat{p}_k \in \mathbf{P}\}$
$l = |\hat{\mathbf{P}}|$
if $l < \mathbb{T}$ **then**
    $\hat{\mathbf{P}} = knn(\mathbf{P}, \mu, l)$ # Calculate the l nearest neighbors of μ in $\mathbf{P}$
    $l = \mathbb{T}$
end if
$p = \frac{1}{l}\sum_{j=1}^{l}\hat{p}_j$

word tune, the miscellaneous tune, the crying tune, and numerous minor tunes. Each tune has a unique rhythmic style. The phoneme set we use contains 93 phonemes (including the special phoneme *sp*). Our dataset contains 79 of them. Phonemes that are not included are rarely used in real life. According to MIDI standard[5], our dataset has a pitch range of 43 (G2,98Hz) -78 (F#5,740Hz).

Method

Model

Our Gezi Opera synthesis system comprises two cascaded models: the acoustic model that generates intermediate acoustic representation (such as mel-spectrogram) from input and the vocoder that generates audio from acoustic representation. We use a HifiGAN (Kong et al. 2020) as our vocoder which is specially designed for SVS tasks [6].

Acoustic Model The structure of our acoustic model is shown in Figure 1. The generator of the GAN can be seen as a conditional image generator composed of a conditioner integrating various conditional inputs and a mel predictor synthesizing the corresponding mel-spectrogram according to the conditions. On this basis, we add a set of discriminators for adversarial training.

Conditioner The input to the conditioner contains four elements: pitch sequence $\mathcal{P}$, phoneme sequence $\mathcal{T}$, duration sequence $\mathcal{D}$, and a speaker label $\mathcal{S}$. The phoneme encoder Enc_t in the conditioner is used to encode the contextual information of the phoneme sequence, which is an FFT block. There are complex slurs in opera pitch sequences, which pose significantly different contextual information between pitch and phoneme sequences. So we add a separate FFT block Enc_p to encode pitch information. On this basis, we employ a Film (Perez et al. 2018) module to better integrate pitch and phoneme information. To enable the synthesis of opera for a specific speaker, that is, to model information related to the speaker, such as timbre and singing style, we use a separate speaker label $\mathcal{S}$. The conditioner outputs a unified conditional representation $\mathcal{C}$, then fed into the mel predictor. We add a variance adapter to the conditioner to model information such as energy following FastSpeech2 (Ren et al. 2020).

[5]https://midi.org/

[6]https://github.com/MoonInTheRiver/DiffSinger/blob/master/docs/README-SVS-popcs.md

Mel Predictor The unified conditional representation $\mathcal{C}$ has a considerable length which can vary a lot (usually from a few hundred to several thousand). In this case, we need not only to model the long-range dependencies of the sequence but also to consider the strong local correlation and translation invariance in opera audio. We employ Conformer (Gulati et al. 2020) as the skeleton of our mel predictor for the following reasons: 1) The multi-head self-attention module in Conformer can effectively model long-range dependencies, which is crucial for modeling long vowel pronunciation in Gezi Opera. 2) Convolutional layers are employed in the Conformer, and the convolutional layers' inductive preference can effectively capture local relevant information. 3) Conformer employs relative position encoding, which can better model translation invariance than absolute position encoding.

Discriminator Our discriminator architecture is borrowed from PatchGAN (Isola et al. 2017) which can better control the generation of each specific small block in the mel-spectrogram. We also provide the unified conditional representation $\mathcal{C}$ to the discriminator to enable better discrimination (Wu and Luan 2020). In response to the problem of significant duration variance in Gezi Opera, we introduce the multi-length GAN (ML-GAN) from HifiSinger (Chen et al. 2020) which employs a set of discriminators to process mel-spectrograms of different lengths. Inspired by the sub-frequency GAN (SF-GAN) in HifiSinger, we design a modified SF-GAN to model the mel-spectrogram in different frequency domains to improve the fullness of the pitch range in the generated audio, making the audio more expressive. We use three parallel discriminators in SF-GAN to process the low, medium, and high-frequency regions of the mel-spectrogram. We assign different weights to different discriminators as there are significant differences in the characteristic and information distribution of the mel-spectrogram along the frequency dimension. Different discriminators can focus on specific patterns in different frequency regions of the mel-spectrogram while the importance of these patterns decreases from low-frequency region to high-frequency region. For ML-GAN, we do not use randomly sampled fragments of different lengths as input to the discriminator. Instead, we segment the spectrogram to a specific length and stack them before sending them to different discriminators for more effective use of data.

Losses

Some previous works, such as Fastspeech (Ren et al. 2019) and Tactron (Wang et al. 2017), only used L1 loss or L2 loss as reconstruction loss. However, L1 loss or L2 loss can cause blurring in the generated spectrogram (Cho et al. 2022; Hono et al. 2019; Zhang et al. 2023), as they are only good at modeling low-frequency structural information (represented by the position and shape of each harmonic on the spectrogram), but have difficulty to model high-frequency detail information (represented by harmonic jitter and clear boundaries on the spectrogram). To address this issue, we employ adversarial losses L_G, L_D:

$$\begin{aligned} L_G =& \mathbb{E}_{\hat{y}}[\sum_{i\in\{l,m,h\}} \rho_i(1 - D_i(\hat{y},\mathcal{C})) \\ &+ \sum_{t\in\{s_1,s_2,s_3\}} \frac{1}{3}(1 - D_t(\hat{y},\mathcal{C}))] \end{aligned} \tag{1}$$

$$\begin{aligned} L_D =& \mathbb{E}_{y,\hat{y}}[\sum_{i\in\{l,m,h\}} \rho_i(1 - D_i(y,\mathcal{C}) + D_i(\hat{y},\mathcal{C})) \\ &+ \sum_{t\in\{s_1,s_2,s_3\}} \frac{1}{3}(1 - D_t(y,\mathcal{C}) + D_t(\hat{y},\mathcal{C}))] \end{aligned} \tag{2}$$

$$\hat{y} = G(\mathcal{P},\mathcal{T},\mathcal{S},\mathcal{D}) \tag{3}$$

where D_l, D_m, and D_h are three SF-GAN discriminators that handle the low, medium, and high-frequency regions of the spectrogram, respectively. ρ_l, ρ_m, and ρ_h are the corresponding weights. Discriminators D_{s_1}, D_{s_2}, and D_{s_3} are three ML-GAN discriminators that handle three different lengths of segments of the spectrogram. $\mathcal{C}$ is the output of the conditioner. We add another feature matching loss L_{fm} to assist the training (Chen et al. 2020):

$$\begin{aligned} L_{fm} =& \mathbb{E}_{y,\hat{y}}[\frac{1}{N}\sum_{i\in\{l,m,h\}}\sum_{l\in[1,N]} \rho_i \|D_i^l(y,\mathcal{C}) - D_i^l(\hat{y},\mathcal{C})\|_2^2 \\ &+ \frac{1}{N}\sum_{t\in\{s_1,s_2,s_3\}}\sum_{l\in[1,N]} \frac{1}{3}\|D_t^l(y,\mathcal{C}) - D_t^l(\hat{y},\mathcal{C})\|_2^2] \end{aligned} \tag{4}$$

where D_i^l (D_t^l) represents the normalization feature of the l-th of N intermediate layers in the i-th (t-th) discriminator.

LPIPS (Zhang et al. 2018) is a distance metric often used in image generation to replace or supplement L1 loss and L2 loss as it can balance low-dimensional perceptual features with high-dimensional semantic features. Therefore, we introduce this distance to supplement the L1 reconstruction loss. To calculate this loss, we first use FastSpeech2 to generate fuzzy mel-spectrograms, and then train a convolutional binary classification network C to distinguish them from the real mel-spectrograms. The perceptual loss L_{per} can be calculated as:

$$L_{per} = \mathbb{E}_{y,\hat{y}}[\frac{1}{L}\sum_{l\in[1,L]} \|C^l(y) - C^l(\hat{y})\|_2^2] \tag{5}$$

where L represents the number of intermediate layers in the classification network, and C^l represents the normalization feature of the l-th intermediate layer in the network.

To prevent the generator loss L_G from becoming too large and causing unstable training, during the training process, we adapt L_G by multiplying it with an adaptive weight ω:

$$\omega = \frac{\|d(L_1 + L_{per})/dw\|_2^2}{\|d(L_G)/dw\|_2^2 + \epsilon} \tag{6}$$

where w represents the parameters of the last layer in the generator and ϵ is a positive value close to zero.

Finally, the total Loss is expressed as L_{total}:

$$L_{total} = L_1 + \omega L_G + L_D + \alpha_p L_{per} + \alpha_f L_{fm} \tag{7}$$

where α_p, α_f represent different loss weights.

Speech Pre-Training

Since the recording audio quality is good enough, coupled with accurate alignment annotation, the data from the Gezi Opera dataset is enough to train a model with high performance. However, if more data can be used for training, the model's performance will improve further. Unfortunately, constrained by the cost of time and labor, we have not been able to collect more Gezi Opera data. However, speech, singing, and Chinese opera all have similar spectral structures, except that the rhythm and melody are different. In other words, we can regard Hokkien speech as Gezi Opera, and use them to help train our model. We crawl 3.77 hours of high-quality Hokkien speech pieces and their corresponding transcripts from the Internet. After MFA alignment, we use Algorithm 1 to extract their pseudo pitch. The processed speech data and Gezi Opera data have the same data structure, that is, they both contain four parts: $\mathcal{P}, \mathcal{T}, \mathcal{S}, \mathcal{D}$. We first use the speech data to train FT-GAN, and then Gezi Opera data for further training.

Experiments

Experiment Settings

Hyperparameter Settings The audio sampling rate is 24kHz (We leave 48kHz sampling rate opera synthesis to future work). On this basis, the window size of the fast Fourier transform is set to 512, and the hop size is set to 128. The spectrogram after fast Fourier transform is converted to a mel-spectrogram of 80 bins. All hidden layer sizes are set to 256. The FFT block in the pitch and phoneme encoder has 4 layers and 8 heads, the same as the Conformer in the mel predictor. The convolutional kernel size of the convolutional layer in the Conformer layer is set to 31. Both SF-GAN and ML-GAN have 3 discriminators and have the same structure: a 3-layer 2d convolutional network with a kernel size of 5. The weights of the three discriminators in SF-GAN are $\rho_l = 0.5, \rho_m = 0.3, \rho_h = 0.2$, the weights of different loss are $\alpha_p = 0.1, \alpha_f = 0.1$. The Learning rate is set to 0.0002, which decays at an exponential decay rate of 0.93 every 10k steps. The mini-batch size used for training in each step is 9.6k audio frames. FT-GAN is first trained for 100k steps on Hokkien speech data and then trained for 200k steps on Gezi Opera data. All training is conducted on one RTX 3090 GPU. More detailed parameter settings are illustrated in our codes.

Baseline Models We reproduce the following baseline models for our comparative experiments:

- **FastSpeech2** (Ren et al. 2020). Although it is a speech synthesis model, employing it for singing voice synthesis tasks can achieve good results (Zhuang et al. 2021; Zhang et al. 2022b). We use the same parameter settings as FT-GAN, with the encoder and the decoder being 4-layer 8-head Feed-Forward Transformers with a hidden size of 256. The model is trained for 300k steps using L1 loss in a mini-batch size of 9.6k audio frames.
- **DiffSinger** (Liu et al. 2022). This is a diffusion singing voice synthesis model, which has good performance in Chinese singing voice synthesis tasks. It mainly composes of a FastSpeech2 and a noise estimation model. For the FastSpeech2 part, we use the same parameter settings as above. The noise estimation model is a 20-layer Wavenet (Oord et al. 2016), which is consistent with the model in the code[7]. The model is trained for 300k steps with a mini-batch size of 9.6k audio frames.
- **Visinger2** (Zhang et al. 2022c) Visinger2 is modified from the speech synthesis model VITS (Kim et al. 2021). It introduces DDSP (Engel et al. 2020) in generating audio to improve performance. We use the official open-source code[8] to reproduce the model. We revise the data process part and train the model with a mini-batch size of 8 on our dataset for 300k steps.

Model	Pro	Nat	Exp
GT	4.54 ± 0.03	4.30 ± 0.03	4.44 ± 0.03
FastSpeech2	4.26 ± 0.04	3.54 ± 0.04	3.76 ± 0.04
DiffSinger	3.96 ± 0.04	3.26 ± 0.04	3.52 ± 0.04
Visinger2	4.44 ± 0.03	3.90 ± 0.04	4.14 ± 0.04
FT-GAN	4.40 ± 0.03	$\mathbf{4.18 \pm 0.03}$	$\mathbf{4.22 \pm 0.04}$
-w/o SP	$\mathbf{4.46 \pm 0.03}$	4.08 ± 0.03	4.20 ± 0.03

Table 2: MOS test result with 95% confidence interval of the ground truth and different models on Gezi Opera synthesis. Pro indicates pronunciation, nat indicates naturalness, exp indicates expressiveness, and SP indicates speech pre-training.

Experimental Results

We use three mean opinion scores (MOS) related to audio quality as our evaluating metrics: 1) pronunciation, the accuracy of lyrics pronunciation in audio; 2) naturalness, whether the singing is as natural as human singing, rather than as stiff as machine-synthesized sounds; 3) expressiveness, whether the melody of the singing is pleasant and emotional. To ensure the reliability of the results, we invite professional Gezi Opera actors and Gezi Opera amateurs to our test.

The results are illustrated in Table 2. FT-GAN outperforms strong baselines except that the pronunciation score is slightly lower than Visinger2 (0.04 MOS), the state-of-the-art model. Especially regarding the naturalness score, FT-GAN exceeds all baselines by a large margin (0.28 MOS improvement compared with the strongest baseline, Visinger2). FT-GAN and all the baselines employ FFT block as the phoneme encoder, which explains why the pronunciation score gap is not significant. However, we add an extra pitch encoder to FT-GAN to better model fine-grained opera pitch information and adopt ML-GAN and SF-GAN from HiFiSinger to improve the training. Together, they make major contributions to enhancing the naturalness of the synthesized opera audio.

[7]https://github.com/MoonInTheRiver/DiffSinger
[8]https://github.com/zhangyongmao/VISinger2

Method	Pro	Nat	Exp	Con
w/o Vowel Morphing	-0.43	-0.36	-0.23	-0.18
w/o Vocal Run	-0.24	-0.39	-0.27	-0.32
w/o Pitch Encoder	-0.20	-0.34	-0.37	-0.25
w/o Discriminator	-0.23	-0.23	-0.11	+0.08
Conformer2FFT	-0.26	-0.32	-0.39	-0.26

Table 3: CMOS (Comparative Mean Opinion Score) test results of the ablation studies on annotation strategies and model structures. Conformer2FFT indicates replacing Conformer with FFT block in the mel predictor and con indicates consistency.

Model	Pro	Nat	Exp
GT	4.53 ± 0.04	4.27 ± 0.06	4.30 ± 0.06
FastSpeech2	3.93 ± 0.05	3.30 ± 0.05	3.50 ± 0.05
DiffSinger	3.37 ± 0.05	2.90 ± 0.05	2.97 ± 0.05
DurIAN	2.97 ± 0.06	2.47 ± 0.05	2.20 ± 0.04
FT-GAN	$\mathbf{4.10 \pm 0.05}$	$\mathbf{3.70 \pm 0.06}$	$\mathbf{3.67 \pm 0.05}$

Table 4: MOS test result with 95% confidence interval of the ground truth and different models on Peking Opera synthesis.

Extensive Experiments

The Effectiveness of Speech Pre-Training To verify the effectiveness of our proposed speech pre-training strategy, we conduct an ablation experiment. We keep all experimental parameters unchanged and only train FT-GAN for 300k steps on the Gezi Opera dataset. The experimental results are listed in the last row of Table 2. It is illustrated that the strategy mainly improves the ability to synthesize more natural opera of FT-GAN with the cost of a slight loss in pronunciation accuracy. We attribute the decline in the model's pronunciation performance to the pronunciation differences between certain characters in Hokkien and Gezi Opera.

Ablation Studies of Annotation Strategies We construct datasets that remove vowel morphing annotation and vocal run annotation respectively and test the synthesis performance of the models trained with them. To verify that vocal run annotation can help the model better model pitch, we add a new metric, consistency, that measures the consistency between real audio pitch and generated audio pitch. The results are illustrated in the first two rows of Table 3. The pronunciation score drops significantly without vowel morphing annotation because this annotation can better reflect pronunciation changes in Chinese operas. The vocal run annotation implies diverse pitch variations help the model reconstruct the pitch curve more accurately, which is illustrated in the consistency score.

Ablation Studies of Model Structures We conduct ablation studies on three key structures in our model, i.e., removing the pitch encoder, removing the discriminator, and replacing Conformer with FFT block in the mel predictor, respectively. We keep the same generator size in all ablation studies to ensure a fair comparison. The results are illustrated in the last three rows of Table 3. All three modifications contribute significantly to model performance, especially the pitch encoder and Confomer, in naturalness and expressiveness scores. An extra pitch encoder and Conformer both help model diverse pitch shifts in Chinese operas, thus they improve the naturalness and expressiveness concerning pitch change reconstruction.

When we deploy the model to the real-world usage scenario, we find that the model trained with speech pre-training strategy has higher robustness to unseen inputs. For example, when users input pitch sequences that do not conform to the rhythm rules of Gezi Opera, the synthesized audio contains fewer noise and pronunciation errors.

Peking Opera Synthesis Performance To verify the performance of FT-GAN in synthesizing other operas and demonstrate its universality, we train FT-GAN on the Peking Opera dataset (Gong et al. 2017). We keep the parameter settings unchanged and train the model for 250k steps, enough for converging. Initially, we train FT-GAN and all the baselines with note input, but all the models fail to synthesize satisfactory audio due to the scarce data. We then change the note input to ground-truth f0 input. All baselines except Visinger2 are trained for 250k steps with unchanged parameter settings. We omit Visinger2 because the official implementation does not support ground-truth f0 input. As a supplement, we incorporate results from DurIAN (Wu et al. 2020) to our test.

The results are shown in Table 4. It is noteworthy that due to the lack of training data (total duration of 1.7 hours), the scores of all models are relatively low. FT-GAN outperforms all baselines by a large margin in all metrics, especially in naturalness (0.4 MOS gain compared with FastSpeech2, the strongest baseline), which is similar to the results in the main experiment. It further demonstrates that FT-GAN is good at synthesizing natural opera.

Conclusion

To push forward research on Chinese opera synthesis, we constructed the first accurately annotated Gezi Opera audio-text alignment dataset. We built a high-performance Gezi Opera synthesis system based on this dataset. We proposed annotation strategies to address the unique problems of Chinese operas. We propose FT-GAN, an acoustic model for fine-grained tune modeling in Chinese opera synthesis. In addition, we proposed a speech pre-training strategy to train the model more fully and improve its inference robustness. However, due to insufficient training data, FT-GAN still encounters issues with electric and mute sounds when the input pitch is extremely high or extremely low or changes between high value and low value frequently.

Acknowledgments

This work is supported by the National Key R&D Program of China (Grant No. 2022ZD0116101), the Key Support Project of NSFC-Liaoning Joint Foundation (Grant No. U1908216), and the Major Scientific Research Project of

the State Language Commission in the 13th Five-Year Plan (Grant No. WT135-38).

References

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

Cho, Y.-P.; Tsao, Y.; Wang, H.-M.; and Liu, Y.-W. 2022. Mandarin Singing Voice Synthesis with Denoising Diffusion Probabilistic Wasserstein GAN. In *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 1956–1963. IEEE.

Engel, J.; Hantrakul, L.; Gu, C.; and Roberts, A. 2020. DDSP: Differentiable digital signal processing. *arXiv preprint arXiv:2001.04643*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144.

Gu, Y.; Yin, X.; Rao, Y.; Wan, Y.; Tang, B.; Zhang, Y.; Chen, J.; Wang, Y.; and Ma, Z. 2021. Bytesing: A chinese singing voice synthesis system using duration allocated encoder-decoder acoustic models and wavernn vocoders. In *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 1–5. IEEE.

Gulati, A.; Qin, J.; Chiu, C.-C.; Parmar, N.; Zhang, Y.; Yu, J.; Han, W.; Wang, S.; Zhang, Z.; Wu, Y.; et al. 2020. Conformer: Convolution-augmented Transformer for Speech Recognition. *Proc. Interspeech 2020*, 5036–5040.

Hono, Y.; Hashimoto, K.; Nankaku, Y.; and Tokuda, K. 2023. Singing Voice Synthesis Based on a Musical Note Position-Aware Attention Mechanism. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Hono, Y.; Hashimoto, K.; Oura, K.; Nankaku, Y.; and Tokuda, K. 2019. Singing voice synthesis based on generative adversarial networks. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6955–6959. IEEE.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1125–1134.

Kenmochi, H.; and Ohshita, H. 2007. VOCALOID-commercial singing synthesizer based on sample concatenation. In *Interspeech*, volume 2007, 4009–4010.

Kim, S.; Na, K.; Lee, C.; An, J.; and Kim, I. 2022. U-Singer: Multi-Singer Singing Voice Synthesizer that Controls Emotional Intensity. *arXiv preprint arXiv:2203.00931*.

Kim, T.-W.; Kang, M.-S.; and Lee, G.-H. 2022. Adversarial Multi-Task Learning for Disentangling Timbre and Pitch in Singing Voice Synthesis. *arXiv preprint arXiv:2206.11558*.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

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

Lee, J.; Choi, H.-S.; Jeon, C.-B.; Koo, J.; and Lee, K. 2019. Adversarially Trained End-to-End Korean Singing Voice Synthesis System. *Proc. Interspeech 2019*, 2588–2592.

Lee, J.; Choi, H.-S.; Koo, J.; and Lee, K. 2020. 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)*, 7224–7228. IEEE.

Lei, Y.; Yang, S.; Wang, X.; Xie, Q.; Yao, J.; Xie, L.; and Su, D. 2022. UniSyn: An End-to-End Unified Model for Text-to-Speech and Singing Voice Synthesis. *arXiv preprint arXiv:2212.01546*.

Liu, J.; Li, C.; Ren, Y.; Chen, F.; and Zhao, Z. 2022. Diffsinger: Singing voice synthesis via shallow diffusion mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 11020–11028.

Lu, P.; Wu, J.; Luan, J.; Tan, X.; and Zhou, L. 2020. XiaoiceSing: A High-Quality and Integrated Singing Voice Synthesis System. *Proc. Interspeech 2020*, 1306–1310.

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

McAuliffe, M.; Socolof, M.; Mihuc, S.; Wagner, M.; and Sonderegger, M. 2017. Montreal forced aligner: Trainable text-speech alignment using kaldi.

Nishihara, M.; Hono, Y.; Hashimoto, K.; Nankaku, Y.; and Tokuda, K. 2023. Singing voice synthesis based on frame-level sequence-to-sequence models considering vocal timing deviation. *arXiv preprint arXiv:2301.02262*.

Oord, A. v. d.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; and Kavukcuoglu, K. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

Oura, K.; Mase, A.; Yamada, T.; Muto, S.; Nankaku, Y.; and Tokuda, K. 2010. Recent development of the HMM-based singing voice synthesis system—Sinsy. In *Seventh ISCA Workshop on Speech Synthesis*.

Perez, E.; Strub, F.; De Vries, H.; Dumoulin, V.; and Courville, A. 2018. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

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

Ren, Y.; Ruan, Y.; Tan, X.; Qin, T.; Zhao, S.; Zhao, Z.; and Liu, T.-Y. 2019. Fastspeech: Fast, robust and controllable text to speech. *Advances in neural information processing systems*, 32.

Saino, K.; Zen, H.; Nankaku, Y.; Lee, A.; and Tokuda, K. 2006. An HMM-based singing voice synthesis system. In *Ninth International Conference on Spoken Language Processing*.

Shen, K.; Ju, Z.; Tan, X.; Liu, Y.; Leng, Y.; He, L.; Qin, T.; Zhao, S.; and Bian, J. 2023. Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers. *arXiv preprint arXiv:2304.09116*.

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

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, T.; Fu, R.; Yi, J.; Wen, Z.; and Tao, J. 2022a. Singing-Tacotron: Global duration control attention and dynamic filter for end-to-end singing voice synthesis. In *Proceedings of the 1st International Workshop on Deepfake Detection for Audio Multimedia*, 53–59.

Wang, Y.; Skerry-Ryan, R.; Stanton, D.; Wu, Y.; Weiss, R. J.; Jaitly, N.; Yang, Z.; Xiao, Y.; Chen, Z.; Bengio, S.; et al. 2017. Tacotron: Towards End-to-End Speech Synthesis. *Proc. Interspeech 2017*, 4006–4010.

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

Wilkins, J.; Seetharaman, P.; Wahl, A.; and Pardo, B. 2018. VocalSet: A Singing Voice Dataset. In *ISMIR*, 468–474.

Wu, J.; and Luan, J. 2020. Adversarially trained multi-singer sequence-to-sequence singing synthesizer. *arXiv preprint arXiv:2006.10317*.

Wu, S.; and Shi, Z. 2023. RealSinger: Ultra-Realistic Singing Voice Generation via Stochastic Differential Equations.

Wu, Y.; Li, S.; Yu, C.; Lu, H.; Weng, C.; Zhang, L.; and Yu, D. 2019. Synthesising expressiveness in peking opera via duration informed attention network. *arXiv preprint arXiv:1912.12010*.

Wu, Y.; Li, S.; Yu, C.; Lu, H.; Weng, C.; Zhang, L.; and Yu, D. 2020. Peking opera synthesis via duration informed attention network. *arXiv preprint arXiv:2008.03029*.

Yang, F.-R.; Cho, Y.-P.; Yang, Y.-H.; Wu, D.-Y.; Wu, S.-H.; and Liu, Y.-W. 2021. Mandarin singing voice synthesis with a phonology-based duration model. In *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 1975–1981. IEEE.

Ye, Z.; Xue, W.; Tan, X.; Chen, J.; Liu, Q.; and Guo, Y. 2023. CoMoSpeech: One-Step Speech and Singing Voice Synthesis via Consistency Model. *arXiv preprint arXiv:2305.06908*.

Yi, Y.-H.; Ai, Y.; Ling, Z.-H.; and Dai, L.-R. 2019. Singing Voice Synthesis Using Deep Autoregressive Neural Networks for Acoustic Modeling. *Proc. Interspeech 2019*, 2593–2597.

Yu, C.; Lu, H.; Hu, N.; Yu, M.; Weng, C.; Xu, K.; Liu, P.; Tuo, D.; Kang, S.; Lei, G.; et al. 2019. Durian: Duration informed attention network for multimodal synthesis. *arXiv preprint arXiv:1909.01700*.

Zhang, L.; Li, R.; Wang, S.; Deng, L.; Liu, J.; Ren, Y.; He, J.; Huang, R.; Zhu, J.; Chen, X.; et al. 2022a. M4Singer: A Multi-Style, Multi-Singer and Musical Score Provided Mandarin Singing Corpus. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595.

Zhang, Y.; Cong, J.; Xue, H.; Xie, L.; Zhu, P.; and Bi, M. 2022b. 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)*, 7237–7241. IEEE.

Zhang, Y.; Xue, H.; Li, H.; Xie, L.; Guo, T.; Zhang, R.; and Gong, C. 2022c. VISinger 2: High-Fidelity End-to-End Singing Voice Synthesis Enhanced by Digital Signal Processing Synthesizer. *arXiv preprint arXiv:2211.02903*.

Zhang, Z.; Zheng, Y.; Li, X.; and Lu, L. 2022d. WeSinger: Data-augmented Singing Voice Synthesis with Auxiliary Losses. *arXiv preprint arXiv:2203.10750*.

Zhang, Z.; Zheng, Y.; Li, X.; and Lu, L. 2023. WeSinger 2: fully parallel singing voice synthesis via multi-singer conditional adversarial training. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

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