Multi-Singer: Fast Multi-Singer Singing Voice Vocoder With A Large-Scale Corpus

Rongjie Huang, Feiyang Chen, Yi Ren, Jinglin Liu, Chenye Cui, Zhou Zhao*
{rongjiehuang,chenfeiyang,rayeren,jinglinliu,chenyecui,zhaozhou}@zju.edu.cn
Zhejiang University

ABSTRACT

High-fidelity multi-singer singing voice synthesis is challenging for neural vocoder due to the singing voice data shortage, limited singer generalization, and large computational cost. Existing open corpora could not meet requirements for high-fidelity singing voice synthesis because of the scale and quality weaknesses. Previous vocoders have difficulty in multi-singer modeling, and a distinct degradation emerges when conducting unseen singer singing voice generation. To accelerate singing voice researches in the community, we release a large-scale, multi-singer Chinese singing voice dataset OpenSinger. To tackle the difficulty in unseen singer modeling, we propose Multi-Singer, a fast multi-singer vocoder with generative adversarial networks. Specifically, 1) Multi-Singer uses a multi-band generator to speed up both training and inference procedure. 2) to capture and rebuild singer identity from the acoustic feature (i.e., mel-spectrogram), Multi-Singer adopts a singer conditional discriminator and conditional adversarial training objective. 3) to supervise the reconstruction of singer identity in the spectrum envelopes in frequency domain, we propose an auxiliary singer perceptual loss. The joint training approach effectively works in GANs for multi-singer voices modeling. Experimental results verify the effectiveness of OpenSinger and show that Multi-Singer improves unseen singer singing voices modeling in both speed and quality over previous methods. The further experiment proves that combined with FastSpeech 2 as the acoustic model, Multi-Singer achieves strong robustness in the multi-singer singing voice synthesis pipeline.

CCS CONCEPTS

• **Applied computing** → **Sound and music computing**; • **Computing methodologies** → **Natural language generation**.

KEYWORDS

singing voice synthesis; singing voice corpus; multi-singer modeling; generative adversarial network

*Corresponding author.

MM '21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475437

ACM Reference Format:
Rongjie Huang, Feiyang Chen, Yi Ren, Jinglin Liu, Chenye Cui, Zhou Zhao. 2021. Multi-Singer: Fast Multi-Singer Singing Voice Vocoder With A Large-Scale Corpus. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3474085.3475437

1 INTRODUCTION

Singing voice synthesis (SVS) aims to synthesize high-quality and expressive singing voices based on musical score information. Singing voice synthesis (SVS) systems [2, 14, 22] take music score and lyric information as input to generate singing voices, and these systems have been widely deployed in music softwares, music boxes, and so on. SVS systems could generate singing voices with comparable quality to reference songs, which attract widespread research interest.

Following the essential components similar to TTS systems, SVS systems generally adopt an acoustic model [5, 23] to convert the musical scores into acoustic features, and a vocoder [2, 21] to generate audio waveform from acoustic features. Neural vocoders can synthesize natural-sounding speech, which generally determines the upper bound of generated sound quality. In this paper, we concentrate on waveform modeling in vocoder.

Unlike traditional TTS [3, 32–34, 44], there are several challenges to build a multi-singer SVS system: 1) Open source and high-quality singing voice data. Unlike speech, high-quality singing voices are commonly produced by professional singers. Because of the high cost of recording and labeling songs, researchers hardly have access to large and open-source singing voice corpora. 2) Fast audio synthesis against limited computation resource. For application deployment, generation speed and computational cost need further consideration. 3) Multi-singer modeling. Timbres could be wildly different among groups while singing voices vary from expression and style. When applying SVS systems for unseen singer modeling, there comes distinct degradation in synthetic singing voice quality.

In the past few years, researchers work to address the challenges above in singing voices modeling, while some problems emerge: 1) SVS systems like DeepSinger [35] mine data from the web, but processed data with noise still could not meet requirements for high-fidelity SVS synthesis. Further, although several singing voice datasets such as MIR-1K dataset [15] and JukeBox [7] have been released for research purposes, but the corpora are not so large as expected for multiple tasks. 2) Several parallel generation methods [40, 42] have been proposed to speed up waveforms synthesis. However, existing multi-band architectures do not consider characteristic differences among frequency bands, so a powerful frequency-adapted multi-band technique is required. 3) Researchers

investigate ways to generate high-quality waveforms during multi-singer modeling. Previous multi-speaker data training strategy [8] increases model generalization. Unfortunately, without explicitly adopting architecture for singer identity reconstruction, vocoders would be data-hungry, and generalization restriction still comes. Giving additional information during generation would be another strategy. Speaker Conditional WaveRNN [28] takes speaker embeddings as additional input, while extra embedding would be sometimes hard-earned during inference procedure and slows down generation as well.

To accelerate SVS research, we assemble an open-source, large-scale, and multi-singer singing voice corpus OpenSinger. To the best of our knowledge, OpenSinger is the first open-source Chinese singing voice dataset. We have attached part of OpenSinger to the supplementary materials, and we will release the entire dataset after paper publication. To overcome afore problems in this paper, we propose Multi-Singer, which achieves computational efficiency and keeps powerful capability for multi-singer singing voices modeling. To be more specific, 1) we introduce a novel multi-band generator, which speeds up singing voice generation and improves the audio quality of different frequency bands. 2) Then we introduce the singer conditional discriminator with conditional loss function for high quality and similarity multi-singer singing voice generation. 3) To further reconstruct singer representations in the frequency domain, we propose an auxiliary singer perceptual loss based on embeddings extracted from a pre-trained speaker encoder. The proposed training method effectively works in GANs for multi-singer singing voice modeling.

Experimental results show that Multi-Singer can generate high-fidelity multi-singer singing voices and achieve the best mean opinion score (MOS) among existing neural vocoders. Combined with FastSpeech 2 as an acoustic model, Multi-Singer shows strong robustness in singing voice synthesis systems. Multi-Singer is substantially faster than most neural vocoders, and it samples 125 times faster than real-time on single NVIDIA 2080Ti GPU with comparable quality to an autoregressive counterpart.

2 RELATED WORKS

In this section, we overview existing datasets, provide a singing voice synthesis background and briefly review several variations of vocoders.

2.1 Dataset

Training TTS and SVS systems both require a significant amount of annotated data[9, 10, 15]. The rapid increase in the amount of multimedia content on the Internet in recent years makes data much more important. Researchers have released speech and singing voice corpora, varying from languages, speakers, and so on.

2.1.1 *Speech.*

For speech synthesis, various datasets are available for different tasks. LSSED [11] is a challenging large-scale English speech emotion dataset, which has data collected from 820 subjects to simulate real-world distribution. AISHELL-3 [36] contains roughly 85 hours of emotion-neutral recordings spoken by 218 native Chinese speakers, which could be applied for multi-speaker speech synthesis. Data could be collected with different methods, including mining from webs automatically, recording manually, and so on. The read English speech LibriSpeech corpus [27] is derived from audiobooks. Japanese multi-speaker singing-voice corpus JVS-MuSiC [37] is produced in a recording studio, and the recordings were controlled by a professional sound director.

2.1.2 *Singing voices.*

Singing voice data differs greatly from speech. Our preliminary research concludes that the main differences between singing voices and speech lie in phoneme duration and pitch (i.e., fundamental frequency), which we would discuss in appendix A in the supplementary materials. Unlike TTS with sufficient transcribed data, SVS suffers from data shortage due to its high recording and annotation cost and stricter copyright issues in the music domain. Limited singing datasets of different sizes and annotated contents are available for research purposes, and here we introduce a few singing voice datasets for comparison in Table 1.

The MIR-1K dataset [15] establish the first comprehensive and publicly available dataset for singing voice separation, which does not contain segmentation on the word level. The proposal of multi-singer NUS-48E corpus [10] is an ongoing effort toward a comprehensive, well-annotated dataset for singing voice related research. But NUS-48E dataset includes 48 songs with reasonably balanced phoneme distribution, which is not large enough for SVS systems training. JukeBox [7] contains 467 hours of singing audio data sampled at 16 kHz, which has been downloaded from the Internet Archive (IA) with a wide variety of acoustic environments and recording apparatus. Hence, it is not suitable for high-quality singing voice synthesis.

To conclude, the above data could not meet our requirements for singing voice synthesis research in terms of quality and quantity. Here in this paper, we propose an open-source, large-scale, and multi-singer singing voice corpus OpenSinger.

2.2 Singing voice synthesis

Singing voice synthesis (SVS) is a generative task that produces acoustic waveforms of singing given lyrics and music score input. A typical singing voice synthesis system consists of an acoustic model to convert musical scores into acoustic features and a vocoder to generate audio waveforms from acoustic features. Previous works have conducted studies on SVS from multiple aspects. DeepSinger [35] is a multi-lingual SVS system built from scratch using singing training data mined from music websites. Choi at all [6] build a Korean singing voice synthesis system using an autoregressive algorithm that generates spectrogram with the boundary equilibrium GAN objective. Chen at all [2] introduce multi-scale adversarial training in both the acoustic model and vocoder to improve singing modeling. As the papers say, these previous SVS systems could generate natural singing voices. However, because vocoders in such SVS systems are not designed towards multiple singers, there would be a distinct degradation in quality when generating unseen singers' voices.

2.3 Vocoder

With the powerful model assumption and the solid theoretical ground, autoregressive vocoders have dominated singing voice modeling for a long time. WaveRNN [18] is an autoregressive model

Name	Task	Language
NUS-48E corpus	singing voice research	English
MIR-1K dataset	singing voice separation	Chinese
JukeBox	singer recognition	18 different languages
OpenSinger	singing voice synthesis	Chinese

Table 1: Usage and language of datasets.

regularly adopted to synthesize waveform in SVS systems. Non-autoregressive vocoder Parallel WaveNet [25] provides a fast waveform generation method based on a teacher-student framework with probability density distillation. Besides, WaveGlow [29] is another non-autoregressive vocoder, which consumes an enormous computation cost during training. More recently, researchers propose diffusion-based models WaveGrad [4] and DiffWave [20] for waveform generation, which are built on prior work of score matching and diffusion probabilistic models.

To avoid the sample-by-sample causal inference or the use of teacher models, researchers have adopted the idea of the generative adversarial network (GAN) to train neural vocoders. MelGAN [21] is a light non-autoregressive vocoder pioneering based on a generative adversarial network, which is free from distillation. HiFi-GAN [19] consists of small sub-discriminators obtaining specific periodic parts of raw waveforms, achieving higher computational efficiency and sample quality. VocGAN [41] applies the joint conditional and unconditional objective, which is inspired by successful results in high-resolution image synthesis. Although these vocoders could be applied in SVS systems, distinct degradations occur when generalizing those systems to unseen singers.

3 CONSTRUCTION OF OPENSINGER

OpenSinger contains pop songs collected from 93 singers, and singing utterances are saved in wav format, sampled at 24 kHz, and quantized by 16 bits. OpenSinger consists of 50 hours of singing voices recorded in a professional recording studio, including 30 hours from 41 females and 20 hours from 25 males apart from the person-of-interest (POI). Figures in appendix A in the supplementary materials summarize the distribution of pitch, sentence-level duration, and phoneme-level duration of utterances. The major features of OpenSinger include:

- Open source. Lack of data could obstruct the construction of SVS systems, so we release our corpus to accelerate research in the community.
- Large scale. Data-hungry singing voice systems need a significant amount of data in the training process. To our best knowledge, OpenSinger is the most extensive Chinese multi-singer singing voice corpus.
- High quality. Similar to text-to-speech, high-quality audios without noise or background sound are essential for high-fidelity singing voice synthesis. Professional singers and studios both ensure high-quality utterances in OpenSinger.

3.1 Data Collection

Collection Procedure In the data collection procedure, we select Chinese traditional and pop songs, and organize a group of 93 professional singers to record 80 hours of singing voices. The recording takes place in a private recording studio. The songs are saved in wav format, sampled at 24 kHz, and quantized by 16 bits.

Data labeling A professional annotation team annotates the utterances in each song. Each utterance is annotated with the name of the song, singer, and reference text, which is the official Latinized notation for marking Chinese pronunciations. Further, We use open source tools pypinyin[1] to convert Chinese lyrics into phonemes.

3.2 Processing

After data collection, the singing voice corpus still could not meet high-quality singing voice generation requirements for the following aspects: 1) Most vocoders need data in the form of relatively short utterances usually up to a few seconds in length, each with corresponding text during training procedures. 2) Pauses and resulting silences commonly remain in raw songs, which would hurt adversarial stability and bring unnecessary calculations. Our processing procedure consists of several stages to fix the issues we listed above for high-quality singing voice generation, including segmentation, silence trimming, and alignment.

Silence trimming After data division, there still usually be some long-term silences in singing voices. Voice Activation Detection (VAD) has been used to remove silent segments, and thus we detect and discard the non-vocal segments using VAD. During silence trimming, each utterance has been manually verified to discard audio samples that do not contain singing vocals every 100ms. Cutting these silence segments could significantly shorten the audios as exceedingly speeds up the alignment formation.

Segmentation Obviously, singing voices with lengthy audio clips are not suitable for memory-limited GPU computation, so we fragment them into many small files. Following the Lyrics-to-Singing alignment in [35], we split the whole song into aligned lyrics and audio. We segment an audio by the frames that are aligned to the separation marks in raw lyrics, making sure each processed sentence's duration is limited to 0-11 seconds.

Alignment For more precise alignment, we adopt Montreal Forced Aligner[2] and take an orthographic transcription of an audio file and generate a time-aligned version. In Montreal Forced Aligner, an annotation for phonemes is generated by aligning the manually-labeled phone strings of the sung lyrics using training conventional Gaussian Mixture Model (GMM) – Hidden Markov Model (HMM) system. The annotated phonemes support us for broad and preliminary observations about the phoneme-level duration of the corpus.

3.3 Statistics

After the data collection and processing procedure, we check for audio quality and conduct statistical evaluation, including sentence-level and phoneme-level duration distribution, pitch distribution and speaker similarity.

Sentence-Level Duration Distribution We have segmented the songs into sentence-level singing voices during audio processing. We plot the sentence-level duration distribution of the male and female singing voices separately. As shown in appendix A in the supplementary materials, sentence durations between genders are

[1] https://github.com/mozillazg/python-pinyin
[2] https://github.com/MontrealCorpusTools/Montreal-Forced-Aligner

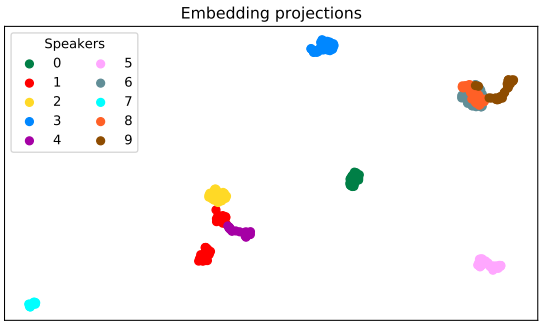


Figure 1: UMAP projection of 10 utterances for each of the 10 singers. Different colors represent different singers.

similar with some small variations due to our manual segmentation in advance.

Phoneme-Level Duration Distribution We align the phoneme sequence to the singing audio frames using MFA mentioned above and visualize distribution in appendix A in the supplementary materials. Phoneme durations of male and female singing voices have a similar distribution, mainly scatters around 10 ms to 500 ms.

Pitch Distribution We extract pitch from the audio using Parselmouth[3] and show statistical distribution. Obviously, due to the diversity of timbre and pronunciation habits among genders, the drawn pitch distribution figures differ greatly. Female singing voices exhibit a larger pitch range, trending toward high frequency more significantly. In other words, there are more high-frequency parts in female singing voices.

Speaker Similarity To learn about speaker identity differences among singers in OpenSinger, we visualize the speaker representations in Figure 1 using Resemblyzer[4]. 10 utterances of 10 randomly sampled speakers are chosen, each is converted into a 256-dimensional embedding and reduced to 2-dimensional with Uniform Manifold Approximation and Projection (UMAP). It has presented a large inter-speaker distance among singers and demonstrated the singer diversity in OpenSinger.

3.4 Speech and Singing voices

Several characteristic differences exist between singing voice and speech generation: 1) Singing voices include richer emotional information, which varies from singing expression and style. 2) Singing voices have longer continuous pronunciations and contain more high-frequency parts. To verify our thinking, we conduct statistical analysis on the Chinese speech dataset CSMSC[5] and plot the phoneme-level duration as well as pitch distribution. Because sentences are cut and segmented manually, we haven't considered sentence-level discrepancies. Table 2 come a few interesting phenomena: 1) Since singing voices are more various and diverse, statistic distributions of singing voices become broader with large variance. 2) Pitch and phoneme-level duration in speech corpus are lower than that in singing voices on average, demonstrating the longer continuous pronunciation and more high-frequency parts in singing voices.

[3]https://github.com/YannickJadoul/Parselmouth
[4]https://github.com/resemble-ai/Resemblyzer
[5]https://github.com/mozillazg/python-pinyin

Item	Pitch (Hz)		Phoneme duration (ms)	
	mean	std	mean	std
Singing Voice (OpenSinger)	280.36	94.56	186.15	143.92
Speech (CSMSC)	250.97	60.63	128.85	78.45

Table 2: Statistics for pitch and phoneme-level duration.

To conclude, the characteristic differences between speech and singing voice we analyze above have brought additional challenges for singing voices synthesis researches, and reduce researchers' enthusiasm to assemble singing voice corpora. To our best knowledge, OpenSinger is the largest open-source, multi-singer, Chinese singing voice corpus, and we hope that the release of OpenSinger could contribute to the community.

4 MULTI-SINGER

4.1 Motivation

4.1.1 Multi-band generartion.

In recent years, state-of-the-art vocoders have significantly improved the singing voice quality and spawned SVS systems deployments. However, due to the high computational cost and time-consumed generation, real-time applications could become challenges. Researchers have adopted multi-band generation such as Multiband-WaveRNN [42], Multiband-MelGAN [40] to speed up waveform modeling. Related multi-band vocoders generate each sub-band of waveform, and then conduct bands splicing using Pseudo Quadrature Mirror Filter Bank (PQMF) [24]. Nevertheless, previous multi-band techniques are designed towards fast generation and lack considering the characteristic differences among frequency bands (e.g., short-period high-frequency band and long-period low-frequency band), usually resulting in the limitation of synthetic singing voice quality. This paper proposes the frequency-adaptive multi-band generation technique to adjust singing voice synthesis in both speed and quality.

4.1.2 Multi-singer modeling.

Sufficient singer generalization means that the vocoder could generate high-fidelity audio in various singer domains, regardless of whether the input has been encountered during training or has come from an out-of-domain singer. Researchers have investigated ways to supervise vocoders for learning singer identity in multi-singer modeling.

Multi-speaker data Training on data of multiple speakers could probably improve model generalization. Using multi-speaker data, researchers [8] demonstrate better performance in speaker similarity for multi-speaker speech synthesis. However, without explicitly adopting architecture for speaker identity reconstruction, vocoders would be data-hungry and encounter generalization restriction. A distinct degradation emerges when we adapt these vocoders to the unseen speakers modeling.

Extra embedding input Taking embedding as additional inputs could be another way to handle singer generalization. Transfer learning from speaker verification to multi-speaker text-to-speech synthesis [17] takes the speaker verification network as the speaker encoder and concatenates the generated speaker embedding to each encoder time step. Speaker Conditional WaveRNN [28] takes

speaker embeddings extracted from a pre-trained speaker verification model as additional input and exploits extra speaker information. However, extra embeddings would sometimes be hard-earned in SVS systems during the inference process. Worse still, it takes a higher computation cost and reduces inference speed intolerably.

To maintain feasibility and improve generalization towards the unseen singers, we have better explore how to teach vocoders explicitly capture singer identity embed in the acoustic feature (i.e., mel-spectrogram) without additional computational cost during singing voice synthesis.

4.2 Overview

Generative adversarial network based vocoder jointly trains a powerful generator G, and convolutional neural network (CNN) discriminator D, to generate time-domain waveform from the corresponding input mel-spectrogram. We have introduced a technique to improve inference speed and quality of synthetic waveform in adaptation to the unseen singer: 1) Firstly, to speed up waveforms modeling, we introduce a multi-band generator, which synthesizes sub-band signals adapted to the frequencies. 2) For multi-singer singing voice generation, Multi-Singer adopts a singer conditional discriminator to judge whether the singer identity of input voices has been properly constructed. 3) For training objective, we introduce joint adversarial training of conditional and unconditional loss and propose singer perceptual loss to penalize the generator for synthesizing singing voices with singer identification bias. The joint training method effectively works in GANs for raw audio generation.

4.3 Generator

To develop high-fidelity waveform generation in parallel, we introduce a multi-band generator. It is acknowledged that waveform signals' characteristics vary in different frequency bands, so we model them separately. The multi-band generator transforms the input noise drawn from a Gaussian distribution to the output waveform in parallel, and the multi-band generation process has been shown in Figure 2. We divide waveform into four frequency bands including two high-frequency bands and two low-frequency bands, and the waveform generation in high and low frequency bands is conducted separately using two distinct frequency-adapted models, respectively. The synthetic sub-band waveforms are merged into the final singing voices through the PQMF filter.

As shown in Figure 2, the generator consists of WaveNet blocks, whose architecture has been discussed in appendix E in the supplementary materials. Specifically, receptive fields and the number of WaveNet blocks vary in these two frequency-adapted models, and they are carefully designed towards different acoustic characteristics among frequency bands. The generator is non-autoregressive and capable of adjusting singing voice synthesis in both speed and quality.

4.4 Discriminator

The architecture of Discriminators has been shown in Figure 2(c), where p1 and p2 denotes the possibility of the sample generated by conditional and unconditional discriminator, respectively. Conditional input has been applied for high-resolution image synthesis tasks and produced successful results [43], and conditional adversarial networks have been widely acknowledged for stable performance in image-to-image translation [16]. Previous vocoder studies have further demonstrated the efficiency of conditional input in discriminators. VocGAN [41] introduces the hierarchically-nested JCU discriminator, which learns intermediate representations directly conditioned on the input mel-spectrogram. GAN-TTS[1] proposes conditional DBlock, where the embedding of the linguistic features are added after the first convolution.

To better supervise the singer identity reconstruction in multi-singer singing voice generation, we adopt a singer conditional discriminator. The singer conditional discriminator judges whether the singer identity of input voices has been properly constructed, and the generator is trained to fool the singer conditional discriminator by increasing the real possibility of the generated sample.

The singer conditional discriminator(p1) first adopts a 256x downsampled block, which is performed using strided average pooling with kernel size 4 and done in 4 stages of 8x, 8x, 2x, and 2x downsampling. After passing through the downsampled block, raw waveforms are converted into 256-dimension vectors. Because singer identifications keep stable in the long-term waveforms, we feed these 256-dimension representations in LSTM layers and obtain steady identity in the final output. We conduct element-wise addition operation between high-level singer identities and reference singer embeddings, and project them to possibility with a linear layer and ReLU activation function.

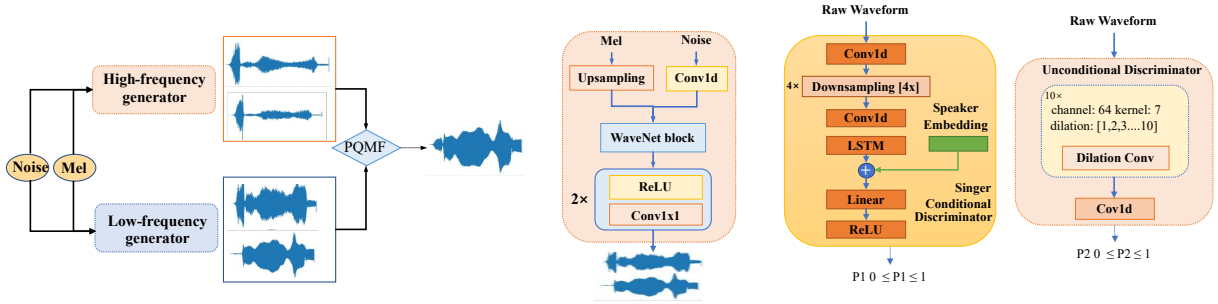
The unconditional discriminator(p2) consists of ten layers of non-causal dilated 1-D convolutions. The strides are set to one and linearly increasing dilations are applied for the 1-D convolutions starting from one to eight except for the first and last layers. Channels and kernel sizes are set to 64 and 5, respectively.

4.5 Training Loss

Training objectives should be carefully designed for stable training and faster convergence. In this section, we first describe **Joint adversarial conditional and unconditional loss**, and then we propose an auxiliary training loss named **Singer Perceptual Loss**. Finally, we adopt **multi-resolution STFT loss** as additional auxiliary loss and decide our final loss function.

Joint adversarial conditional and unconditional (JCU) loss In contrast to conventional adversarial loss, JCU loss combines the conditional and unconditional adversarial losses as Eq. 1. Previous work like VocGAN [41] has demonstrated the efficiency of JCU loss for adversarial training. Here our adversarial conditional loss is significantly different from that in VocGAN. For multi-singer modeling, our proposed conditional loss is conducted by combating between generator and singer conditional discriminator, leading generator to better capture singer identity embed in the acoustic feature input (i.e., mel-spectrogram). The adversarial unconditional loss enhances the generator to synthesize more natural singing voices by classifying ground truth samples to 1 and the synthetic samples to 0. The JCU loss is defined in Eq. 1 and Eq. 2.

$$
\begin{aligned}
L_{adv}(D;G) = \frac{1}{2}\mathbb{E}_x\left[(D(x)-1)^2 + D(y)^2\right] + \\
\frac{1}{2}\mathbb{E}_{x,s}\left[(D(x,s)-1)^2 + D_s(y,s)^2\right],
\end{aligned}
\tag{1}
$$



(a) Multi-band Generation (b) Generator (c) Discriminator

Figure 2: An architecture of Multi-Singer. (a) Multi-band generator which consists of two WaveNet-block-based generators. (b) WaveNet-block-based generator. (c) Singer Conditional and Unconditional Discriminators.

$$L_{adv}(G;D) = \frac{1}{2}\mathbb{E}_y\left[(D(y)-1)^2\right] + \frac{1}{2}\mathbb{E}_{y,s}\left[(D_s(y,s)-1)^2\right], \quad (2)$$

where x, y and s denote the ground truth and synthetic singing voices, and the singer embedding respectively. G is the multi-band generator, D is the unconditional discriminator, D_s is the singer conditional discriminator.

Singer Perceptual Loss In this work, we propose an auxiliary loss function that could improve the vocoder performance for multi-singer singing voice modeling. Perceptual loss is introduced in style reconstruction by Gratys et al. [12] for the first time, and many studies have taken the ideas behind perceptual loss to improve the quality of the outputs generated by a deep-learning-based model. Here we introduce a new loss objective named singer perceptual loss, which enables the generator to sense singer bias and optimize speaker similarity between ground truth and synthetic waveforms in the frequency domain during training.

Singer perceptual loss depends on high-level hidden states extracted from the pre-trained speaker encoder, which we will discuss in appendix C in the supplementary materials. We judge mel-spectrogram in the frequency domain to calculate singer perceptual loss. It is acknowledged that the spectrum envelope of waveforms could provide singer representations, and hence envelope reconstruction in the frequency domain is also the procedure of singer identity reconstruction. Singer perceptual loss supervises singer identification reconstruction more efficiently, and it is defined as follows:

$$L_{spl}(x,y) = \sum_{j=1}^{L}\left(\left\|\phi_j(\mathrm{Mel}(x)) - \phi_j(\mathrm{Mel}(y))\right\|_2\right),$$

where $L_{spl}(x,y)$ denotes singer perceptual loss, $\|\cdot\|_2$ denotes the L2 norms, and let $\phi_j(\mathrm{Mel}(x))$ be the j-th layer hidden state of LSTM in the pre-trained speaker encoder ϕ when processing the mel-spectrogram of signal x.

Multi-resolution STFT Loss To further stabilize the adversarial training process, Multi-Singer adopts a multi-resolution STFT(Short Time Fourier Transform) auxiliary loss. Similar to the previous work [39], we define the STFT loss as follows:

$$L_{m\_sc}(x,y) = \frac{\|\mathrm{STFT}(x) - \mathrm{STFT}(y)\|_F}{\|\mathrm{STFT}(x)\|_F},$$

$$L_{m\_mag}(x,y) = \frac{1}{N}\|\log(\mathrm{STFT}(x)) - \log(\mathrm{STFT}(y))\|_1,$$

where $\|\cdot\|_F$ and $\|\cdot\|_1$ denote the Frobenius and L1 norms; $STFT(\cdot)$ and N denote the STFT magnitude of the m-th STFT parameter set and the number of elements in the magnitude, respectively. L_{m_sc} and L_{m_mag} denote the spectral convergence and log STFT magnitude, respectively.

The final multi-resolution STFT loss is the sum of M losses with different analysis parameters (i.e., FFT size, window size, and hop size), which is represented as follows:

$$L_{stft}(x,y) = \frac{1}{M}\sum_{m=1}^{M}\left(L_{m\_sc}^{(m)}(x,y) + L_{m\_mag}^{(m)}(x,y)\right).$$

Final loss As mentioned above, our auxiliary loss consists of the singer perceptual loss and multi-resolution STFT loss as follows:

$$L_{aux}(G) = \mathbb{E}_{x,y}\left[\frac{1}{2}\left(L_{spl}(x,y) + L_{stft}(x,y)\right)\right],$$

where $L_{stft}(x,y)$ denotes the multi-resolution STFT loss, and $L_{aux}(G)$ denotes the auxiliary loss of generator.

To conclude, our final loss function for the generator is defined as a linear combination of the auxiliary loss, adversarial loss:

$$L_G = L_{aux}(G) + \lambda L_{adv}(G;D),$$

where λ denotes the hyperparameter balancing loss terms and we set $\lambda = 10$. By jointly optimizing the waveform-domain adversarial loss and auxiliary loss including singer perceptual loss and multi-resolution STFT loss, the generator can learn the distribution of the realistic speech waveform effectively.

5 EXPERIMENT

In this section, we first describe the experimental setup including dataset and model configurations. Then we report experimental results and conduct some analyses.

5.1 Experimental Setup

We randomly choose 340 utterances for validation and 60 utterances from 6 singers as the seen singer test set. To evaluate the model generalization to unseen singers, we prepare 5 utterances from each singer including five males and five females as the additional unseen singer test set. The seen singer test set helps judge if the neural vocoder could synthesize high-fidelity singing voices, and

the unseen singer test set helps measure the model's generalization to singer out of domain. At the same time, we take Chinese speech dataset CSMSC for comparison and choose 60 utterances as the test set.

5.2 Model training

Speaker encoder We train the speaker encoder with a few large scale multi-speaker datasets following the guidance in [17]: 1) LibriSpeech Other, which contains 461 hours of speech from a set of 1,166 speakers; 2) VoxCeleb and VoxCeleb2 which contain 139K utterances from 1,211 speakers, and 1.09M utterances from 5,994 speakers, respectively. 3) OpenSinger. We further fine-tune the speaker encoder on the singing voice corpus to ensure it has learned about the distribution of singing voices. Speaker encoder has been trained until convergence (around 50k iterations) as shown in appendix C in the supplementary materials.

Multi-Singer The generator in Multi-Singer takes 80-band mel-spectrograms as input, which are extracted in params (FFT:512, hop size:128, window size:512). At the training stage, a multi-resolution STFT loss would be computed by the sum of three different STFT losses as described in appendix D in the supplementary materials. Multi-Singer is trained for 40k steps with RAdam optimizer to stabilize training. Note that we apply pre-training on the generator for the first 100k steps, after which generator and discriminators are jointly trained. When training previous vocoders such as WaveRNN, MelGAN, and Parallel WaveGAN from scratch, GitHub implementations are used for reproducibility and the configurations follow their original papers.

5.3 Corpus Verification

In order to validate the audio quality of OpenSinger, we train several state-of-the-art neural vocoders such as WaveRNN [18], MelGAN [21] and Parallel WaveGAN [39]. We then conduct evaluations on synthetic seen singer samples[6], and the evaluation matrix has been discussed in appendix B in the supplementary materials.

Our further evaluation lies on the corpus scale. It's well-known that high-fidelity TTS is data-hungry and hence model pre-training is essential in practice. Researchers [8] used to perform the warm-start strategy to overcome data shortage. To explore the effectiveness of additional speech data in SVS training, we pre-train vocoder on a large speech corpus CSMSC (200k steps) and fine-tune the model on OpenSinger until convergence (500k steps).

Corpus verification results have been introduced in Table 3, and we come to the following conclusions:

- OpenSinger simulates the real-world singing voice distribution, and state-of-the-art neural vocoders perform as well as previous papers say. Robust models could be trained using OpenSinger, demonstrating the effectiveness of this dataset.
- A slight performance drop appears when using the speech pre-train strategy, which indicates that additional speech data is not needed since OpenSinger is large enough for high-quality singing voice modeling.

[6]Audio samples are available at https://Multi-Singer.github.io/

Model	Dataset	MOS	FDSD
WaveRNN	Singing Voice	3.59±0.15	0.385
MelGAN	Singing Voice	3.24±0.10	0.864
Parallel WaveGAN	Singing Voice	3.52±0.12	0.484
Parallel WaveGAN	Speech + Singing Voice	3.49±0.10	0.488

Table 3: MOS results with 95% confidence intervals and FDSD for corpus verification.

5.4 Multi-band generation

To verify the effectiveness of the proposed multi-band generator in Multi-Singer, we conduct comparison with competing multi-band architectures such as multiband-MelGAN and multiband-WaveRNN. For fair comparison, we train these models from scratch under the setting in section 5.1 and implement MOS assessments and real-time factor (RTF) evaluation. From the experimental results in Table 4, we draw the following conclusions: 1) Due to the auto-regressive architecture, Multi-band WaveRNN achieves the best performance and generates the most natural sounds, which limits overall generation speed on the other hand. 2) Multi-band MelGAN could achieve fast singing voice synthesis, while a distinct quality degradation in audio comes. 3) as for Multi-Singer, because of the multi-band architecture adapted towards characteristic differences among frequency bands, the non-autoregressive generator could adjust singing voice synthesis in both speed and quality

Model	MOS	RTF
Multi-band MelGAN	3.21±0.10	0.002
Multi-band WaveRNN	3.58±0.13	0.350
Multi-Singer	3.98±0.06	0.008

Table 4: MOS results with 95% confidence intervals and real-time factor (RTF) of each multi-band generation methods.

5.5 Multi-singer modeling

To perform Multi-Singer's better generalization to multiple unseen singers, we evaluate on the synthetic singing voices and compare them with competing architectures. For a fair comparison, we train these vocoders on our proposed multi-singer dataset OpenSinger and use the same experiment setup described in Section 5.1. We implement MOS assessments and present objective evaluations such as Fréchet Deep Speech Distances (FDSD), and speaker cosine similarity as shown in Table 5. We have attached the concrete matrix in appendix B in the supplementary materials. Note that the quality of the added out-of-domain unseen singer recordings is lower than that of seen speaker recordings. Therefore we do not conduct meaningless evaluation across seen and unseen singer test set.

Experimental results represent the robustness of Multi-Singer and its outperform capability of unseen singer modeling. We come to conclusions as follows: 1) Architectures such as MelGAN and Parallel WaveGAN haven't explicitly introduced methods for multi-singer adaptation, thus an unavoidable degradation occurs when modeling singing voices of unseen singers. 2) Singer Conditional WaveRNN (SC-WaveRNN) introduces singer-embedding as additional information to control the singer identity during inferences,

Model	Train FDSD	Seen test singer		Unseen test singer	
		MOS	Cosine similarity	MOS	Cosine similarity
WaveRNN (autoregressive)	0.385	3.53±0.12	0.923	3.59±0.15	0.931
SC-WaveRNN (autoregressive)	0.525	3.59±0.13	0.958	3.63±0.13	0.970
MelGAN	0.864	3.22±0.12	0.964	3.24±0.10	0.943
Parallel WaveGAN	0.484	3.51±0.12	0.934	3.52±0.12	0.944
Multi-Singer	0.412	3.96±0.09	0.959	3.98±0.06	0.967

Table 5: MOS results with 95% confidence intervals, FDSD and cosine similarity for multi-singer modeling. Note that the quality of the added out-of-domain unseen singer recordings is lower than that of seen speaker recordings. Therefore we do not conduct evaluation across seen and unseen singer test set.

Model	Train FDSD	RTF	Seen test singer		Unseen test singer	
			MOS	Cosine similarity	MOS	Cosine similarity
w/o MB-generator	0.432	0.031	3.97±0.07	0.958	4.00±0.08	0.966
w/o SCD	0.445	0.008	3.68±0.09	0.952	3.70±0.08	0.964
w/o SPL	0.461	0.008	3.81±0.10	0.951	3.83±0.09	0.963
Baseline (Multi-Singer)	0.412	0.008	3.96±0.09	0.959	3.98±0.06	0.968

Table 6: MOS results with 95% confidence intervals, FDSD and cosine similarity for ablation study of each component.

while the large computational cost slows down generation and increase the difficulty of further application; and 3) as for Multi-Singer, with the prominent capability to perceive singer identity without extra computation during generation, it adjusts singing voice synthesis in both speed and quality.

5.6 Ablation study

We conduct ablation studies under the settings in Section 5.1 to verify the effectiveness of several components in Multi-Singer, including 1) multi-band generator; 2) singer conditional discriminator; and 3) singer perceptual loss. Table 6 shows the mean opinion score of audio quality as assessed via human listening tests and objective evaluation results. Our analysis leads to the following conclusions: Replacing **multi-band generator** with a full band generator causes a significant decrease in generation speed, and we observe that the quality of the synthetic singing voices drops slightly at the same time. The absence of **Singer Conditional Discriminator (SCD)** results in decreased cosine similarity scores on unseen singers, which suggests that the modified vocoder has difficulties in capturing singer identity. Because abundant singer representations are embedded in the spectrum envelope of singing voices, removing the frequency-domain auxiliary objective **Singer Perceptual Loss (SPL)** could weaken vocoder in the reconstruction of singer representations. Our ablation study shows that the baseline model Multi-Singer can speed up waveforms generation and efficiently reestablish the singer identity of unseen singers in singing voices.

5.7 Singing voice synthesis system

To verify the effectiveness of Multi-Singer in singing voice synthesis systems, we adopt a modified FastSpeech 2 as an acoustic model to convert the words of songs into acoustic features and build an overall system. We have attached the modified FastSpeech 2 in appendix F in the supplementary materials. During training, the configuration follows prior work [31]. Since the F0 and duration are usually known in singing voice synthesis, we remove the pitch and duration prediction, taking the real F0 and phoneme duration as input. FastSpeech 2 converts lyrics, F0, duration and singer embedding into the mel-spectrogram, Multi-Singer converts the mel-spectrogram into singing voices. We perform a MOS scoring test for synthetic audio and get results in Table 7. Experimental results show that Multi-Singer combined with FastSpeech 2 could generate high-quality singing voices and demonstrate the strong robustness of Multi-Singer in the singing voice synthesis system.

Model	MOS
FastSpeech 2 + Multi-Singer	3.95±0.07
Recording	4.03±0.09

Table 7: The MOS results with 95% confidence intervals on each Singing voice synthesis system.

6 CONCLUSION

We released OpenSinger, a large-scale, multi-singer Chinese singing voice dataset. To our best knowledge, OpenSinger was the first Chinese open dataset towards high-fidelity singing voice synthesis, which we hope would accelerate singing voice synthesis research in the community. To speed up waveforms generation and enhance the capability of vocoder in multi-singer modeling, we proposed Multi-Singer, a fast multi-singer singing voice vocoder. We attached singer conditional discriminator and conditional adversarial training objective to improve singer identity reconstruction. To assist Multi-Singer sense singer bias between the synthetic and reference singing voices and supervise singer representations reconstruction, we introduced singer perceptual loss as auxiliary loss function. The corpus evaluation demonstrated the effectiveness of OpenSinger for singing voice synthesis researches. Further experimental results showed that Multi-Singer could speed up the generation and synthesize high-fidelity singing voices of unseen singers. Our further experiments proved that Multi-Singer achieved strong robustness in the singing voice synthesis system. For future work, we will continue to study model generalization to different emotions.

7 ACKNOWLEDGEMENTS

This work was supported in part by the National Key R&D Program of China under Grant No.2020YFC0832505, National Natural Science Foundation of China under Grant No.61836002, No.62072397 and Zhejiang Natural Science Foundation under Grant LR19F020006.

REFERENCES

[1] Mikołaj Bińkowski, Jeff Donahue, Sander Dieleman, Aidan Clark, Erich Elsen, Norman Casagrande, Luis C Cobo, and Karen Simonyan. 2019. High fidelity speech synthesis with adversarial networks. *arXiv preprint arXiv:1909.11646* (2019).

[2] Jiawei Chen, Xu Tan, Jian Luan, Tao Qin, and Tie-Yan Liu. 2020. HiFiSinger: Towards High-Fidelity Neural Singing Voice Synthesis. *arXiv preprint arXiv:2009.01776* (2020).

[3] Mingjian Chen, Xu Tan, Yi Ren, Jin Xu, Hao Sun, Sheng Zhao, Tao Qin, and Tie-Yan Liu. 2020. Multispeech: Multi-speaker text to speech with transformer. *arXiv preprint arXiv:2006.04664* (2020).

[4] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. 2020. WaveGrad: Estimating gradients for waveform generation. *arXiv preprint arXiv:2009.00713* (2020).

[5] S. Choi, W. Kim, S. Park, S. Yong, and J. Nam. 2020. Korean Singing Voice Synthesis Based on Auto-Regressive Boundary Equilibrium Gan. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 7234–7238. https://doi.org/10.1109/ICASSP40776.2020.9053950

[6] Soonbeom Choi, Wonil Kim, Saebyul Park, Sangeon Yong, and Juhan Nam. 2020. Korean singing voice synthesis based on auto-regressive boundary equilibrium gan. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 7234–7238.

[7] Anurag Chowdhury, Austin Cozzo, and Arun Ross. 2020. JukeBox: A Multilingual Singer Recognition Dataset. *arXiv preprint arXiv:2008.03507* (2020).

[8] Erica Cooper, Xin Wang, Yi Zhao, Yusuke Yasuda, and Junichi Yamagishi. 2020. Pretraining Strategies, Waveform Model Choice, and Acoustic Configurations for Multi-Speaker End-to-End Speech Synthesis. *arXiv preprint arXiv:2011.04839* (2020).

[9] Chenye Cui, Yi Ren, Jinglin Liu, Feiyang Chen, Rongjie Huang, Ming Lei, and Zhou Zhao. 2021. EMOVIE: A Mandarin Emotion Speech Dataset with a Simple Emotional Text-to-Speech Model. *arXiv preprint arXiv:2106.09317* (2021).

[10] Zhiyan Duan, Haotian Fang, Bo Li, Khe Chai Sim, and Ye Wang. 2013. 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, 1–9.

[11] Weiquan Fan, Xiangmin Xu, Xiaofen Xing, Weidong Chen, and Dongyan Huang. 2021. LSSED: a large-scale dataset and benchmark for speech emotion recognition. *arXiv preprint arXiv:2102.01754* (2021).

[12] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2015. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576* (2015).

[13] Alexey A. Gritsenko, Tim Salimans, Rianne van den Berg, Jasper Snoek, and Nal Kalchbrenner. 2020. A Spectral Energy Distance for Parallel Speech Synthesis. arXiv:2008.01160 [eess.AS]

[14] Yu Gu, Xiang Yin, Yonghui Rao, Yuan Wan, Benlai Tang, Yang Zhang, Jitong Chen, Yuxuan Wang, and Zejun Ma. 2020. ByteSing: A Chinese Singing Voice Synthesis System Using Duration Allocated Encoder-Decoder Acoustic Models and WaveRNN Vocoders. *arXiv preprint arXiv:2004.11012* (2020).

[15] Chao-Ling Hsu and Jyh-Shing Roger Jang. 2009. On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset. *IEEE Transactions on Audio, Speech, and Language Processing* 18, 2 (2009), 310–319.

[16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1125–1134.

[17] Ye Jia, Yu Zhang, Ron J Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, et al. 2018. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. *arXiv preprint arXiv:1806.04558* (2018).

[18] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron van den Oord, Sander Dieleman, and Koray Kavukcuoglu. 2018. Efficient neural audio synthesis. *arXiv preprint arXiv:1802.08435* (2018).

[19] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis. *arXiv preprint arXiv:2010.05646* (2020).

[20] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. 2020. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761* (2020).

[21] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C Courville. 2019. Melgan: Generative adversarial networks for conditional waveform synthesis. In *Advances in Neural Information Processing Systems*. 14910–14921.

[22] Jinglin Liu, Chengxi Li, Yi Ren, Feiyang Chen, Peng Liu, and Zhou Zhao. 2021. Diffsinger: Diffusion acoustic model for singing voice synthesis. *arXiv preprint arXiv:2105.02446* (2021).

[23] Peiling Lu, Jie Wu, Jian Luan, Xu Tan, and Li Zhou. 2020. XiaoiceSing: A High-Quality and Integrated Singing Voice Synthesis System. arXiv:2006.06261 [eess.AS]

[24] Truong Q Nguyen. 1994. Near-perfect-reconstruction pseudo-QMF banks. *IEEE Transactions on signal processing* 42, 1 (1994), 65–76.

[25] Aaron Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George Driessche, Edward Lockhart, Luis Cobo, Florian Stimberg, et al. 2018. Parallel wavenet: Fast high-fidelity speech synthesis. In *International conference on machine learning*. PMLR, 3918–3926.

[26] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016).

[27] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 5206–5210.

[28] Dipjyoti Paul, Yannis Pantazis, and Yannis Stylianou. 2020. Speaker conditional WaveRNN: Towards universal neural vocoder for unseen speaker and recording conditions. *arXiv preprint arXiv:2008.05289* (2020).

[29] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. 2019. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 3617–3621.

[30] Flavio Protasio Ribeiro, Dinei Florencio, Cha Zhang, and Mike Seltzer. [n.d.]. CROWDMOS: An Approach for Crowdsourcing Mean Opinion Score Studies. In *ICASSP* (2011). IEEE. Edition: ICASSP.

[31] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2020. FastSpeech 2: Fast and High-Quality End-to-End Text to Speech. arXiv:2006.04558 [eess.AS]

[32] Yi Ren, Jinglin Liu, Xu Tan, Chen Zhang, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2020. SimulSpeech: End-to-end simultaneous speech to text translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 3787–3796.

[33] Yi Ren, Jinglin Liu, Xu Tan, Zhou Zhao, Sheng Zhao, and Tie-Yan Liu. 2020. A study of non-autoregressive model for sequence generation. *arXiv preprint arXiv:2004.10454* (2020).

[34] Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2019. Fastspeech: Fast, robust and controllable text to speech. *arXiv preprint arXiv:1905.09263* (2019).

[35] Yi Ren, Xu Tan, Tao Qin, Jian Luan, Zhou Zhao, and Tie-Yan Liu. 2020. Deepsinger: Singing voice synthesis with data mined from the web. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1979–1989.

[36] Yao Shi, Hui Bu, Xin Xu, Shaoji Zhang, and Ming Li. 2020. AISHELL-3: A Multi-speaker Mandarin TTS Corpus and the Baselines. *arXiv preprint arXiv:2010.11567* (2020).

[37] Hiroki Tamaru, Shinnosuke Takamichi, Naoko Tanji, and Hiroshi Saruwatari. 2020. JVS-MuSiC: Japanese multispeaker singing-voice corpus. *arXiv preprint arXiv:2001.07044* (2020).

[38] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. 2018. Generalized end-to-end loss for speaker verification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 4879–4883.

[39] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. 2020. Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6199–6203.

[40] Geng Yang, Shan Yang, Kai Liu, Peng Fang, Wei Chen, and Lei Xie. 2020. Multi-band MelGAN: Faster Waveform Generation for High-Quality Text-to-Speech. arXiv:2005.05106 [cs.SD]

[41] Jinhyeok Yang, Junmo Lee, Youngik Kim, Hoonyoung Cho, and Injung Kim. 2020. VocGAN: A High-Fidelity Real-time Vocoder with a Hierarchically-nested Adversarial Network. *arXiv preprint arXiv:2007.15256* (2020).

[42] Chengzhu Yu, Heng Lu, Na Hu, Meng Yu, Chao Weng, Kun Xu, Peng Liu, Deyi Tuo, Shiyin Kang, Guangzhi Lei, et al. 2019. Durian: Duration informed attention network for multimodal synthesis. *arXiv preprint arXiv:1909.01700* (2019).

[43] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. 2018. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE transactions on pattern analysis and machine intelligence* 41, 8 (2018), 1947–1962.

[44] Kexun Zhang, Yi Ren, Changliang Xu, and Zhou Zhao. 2021. WSRGlow: A Glow-based Waveform Generative Model for Audio Super-Resolution. *arXiv preprint arXiv:2106.08507* (2021).

A STATISTICS

As shown in Github pages[7], there come differences between speech data and singing voice data. To be more specific, 1) Singing voices vary from expression and style, including richer emotional information. 2) Singing voices usually have a high sampling rate, which causes a wider spectrogram band in the frequency domain and more high-frequency parts.

B EVALUATION

B.1 Mean Opinion Scores

All our Mean Opinion Score (MOS) tests are crowdsourced and conducted by native speakers. We refer to the rubric for MOS scores in [30], and the scoring criteria has been included in Table **??** for completeness. The samples are presented and rated one at a time by the testers.

Rating	Naturalness	Definition
1	Bad	Very annoying and objectionable dist.
2	Poor	Annoying but not objectionable dist.
3	Fair	Perceptible and slightly annoying dist
4	Good	Just perceptible but not annoying dist.
5	Excellent	Imperceptible distortions

Table 8: Ratings that have been used in evaluation of speech naturalness of synthetic and ground truth samples.

B.2 Fréchet Deep Speech Distances

Fréchet Deep Speech Distances (FDSD) judges the quality of synthetic audio samples based on their distance to a reference set. These distances are conceptually similar to the FID (Fréchet Inception Distance). The energy distance can be combined with GAN-based learning, further improving on either individual technique. As the paper [13] says, FDSD is a proper scoring rule with respect to the distribution over spectrograms of the generated waveform audio.

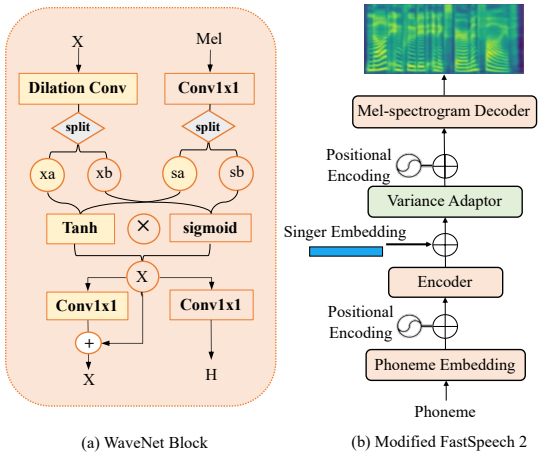
B.3 Cosine similarity

Cosine similarity is an objective metric that measures speaker similarity among multi-singer audio. $\cos\_sim(A, B) = \frac{A \cdot B}{||A|| ||B||}$. We also compute the average cosine similarity between embeddings extracted from synthetic speech and the ground truth embeddings to measure the speaker similarity performance objectively. Embeddings of utterances from the same speaker have high cosine similarity, while those from different speakers are far apart in the embedding space.

C SPEAKER ENCODER

Speaker verification verifies the speaker identity and tells if the representations come from a related speaker. A speaker-discriminative neural encoder [38] on a speaker verification (SV) task using a state-of-the-art generalized end-to-end loss. After training on a large amount of data, the speaker encoder could attain robust representations that capture an ample singer identity space. As a result, speaker encoders are usually used for feature extraction, which effectively captures the audio's long-term speaker identity.

[7]Statistical results are available at https://Multi-Singer.github.io/



(a) WaveNet Block (b) Modified FastSpeech 2

Figure 3: Architectures.

Inspired by previous work that has reported the effectiveness of combining a well-trained d-vector model with a TTS model, we build a speaker encoder that projects the mel-spectrogram from the speech utterance to a 256-dimensional. Generalized end-to-end loss (GE2E) makes the training of speaker verification models more efficient than the previous tuple-based end-to-end (TE2E) loss function. During the training process as shown in appendix C in the supplementary materials, the speaker encoder could reduce intra-speaker variance and separate different speakers apart.

D MULTI-RESOLUTION STFT LOSS DETAILS

Here we introduce details of the multi-resolution STFT loss.

FFT size	Frame shift	Window size
1024	600	120
2048	120	250
512	240	50

Table 9: The details of the multi-resolution STFT loss. A hanning window was applied before the FFT process.

E WAVENET BLOCK

The architecture of WaveNet block [26] in generator has been shown in Figure 3. X and Mel denote noise and mel-spectrogram, respectively. Noise passes through the dilated convolution layers, and X, Mel are divided into xa, xb and sa, sb, respectively. After the sigmoid-tanh calculation, the processed feature passes through two fully-connected networks and output H and X, which would be fed into the next dilated convolution layer.

F MODIFIED FASTSPEECH 2

As shown in Figure 3, our modified FastSpeech 2 converts lyrics, F0, duration, and singer embedding into the mel-spectrogram, after which Multi-Singer converts the mel-spectrogram into singing voices.