SUSing: SU-net for Singing Voice Synthesis

Xulong Zhang, Jianzong Wang*, Ning Cheng, Jing Xiao
Ping An Technology (Shenzhen) Co., Ltd., China

Abstract—Singing voice synthesis is a generative task that involves multi-dimensional control of the singing model, including lyrics, pitch, and duration, and includes the timbre of the singer and singing skills such as vibrato. In this paper, we proposed SU-net for singing voice synthesis named SUSing. Synthesizing singing voice is treated as a translation task between lyrics and music score and spectrum. The lyrics and music score information is encoded into a two-dimensional feature representation through the convolution layer. The two-dimensional feature and its frequency spectrum are mapped to the target spectrum in an autoregressive manner through a SU-net network. Within the SU-net the stripe pooling method is used to replace the alternate global pooling method to learn the vertical frequency relationship in the spectrum and the changes of frequency in the time domain. The experimental results on the public dataset Kiritan show that the proposed method can synthesize more natural singing voices.

Index Terms—Singing voice synthesis, SU-net, Stripe pooling, Convolution

I. INTRODUCTION

Singing voice synthesis (SVS) generates a singing sound with musical characteristics according to a given lyrics or music score. The task of singing voice synthesis is similar to the text-to-speech (TTS) in speech processing, and the synthesis speech is generated according to the given text. With the development of text-to-speech technology, many technologies [1]–[7] have been successfully applied to the task of singing voice synthesis. Both of the tasks of TTS and SVS encode the lyrics or text into an acoustic variable, through the convolution process to draw attention to frequency bins dependence on the objective function and used an autoregressive method to perform end-to-end training on frame-level input. In the decoder, the RNN uses the current encoder output and the Mel spectrum of the previous time sequence as input to predict the Mel spectrum of the current time sequence. Lu et al. [31] added F0 prediction and duration prediction for singing voice synthesis based on phoneme modeling based on the FastSpeech model in TTS. Choi et al. [32] used a GAN-based singing synthesis method combined with boundary balance as the objective function and used an autoregressive method to generate a spectrum to solve the time continuity between two adjacent outputs.

In this paper, we propose an end-to-end singing voice synthesis method with a Striped U-net (SU-net). The proposed method is motivated by the successfully adapted U-net [33] on the singing voice separation task, which maps the spectrum to the mask of the target spectrum. In the proposed method, the singing voice synthesis task is regarded as the translation task from the encoded embedding of the musical score to the spectrum. Regarding the relationship between the fundamental frequency and the harmonics composition of the singing voice in the spectrum. We use stripe pooling during the convolution process to draw attention to frequency bins dependence on alignment [15]–[17], and the synthesis method based on statistical parameters [18], [19], to the present end-to-end deep learning method [1], [20]–[23]. The concatenative based synthesis method has high sound quality but weak generalization ability and phonemes that are not covered in the training set cannot be generated [24]. The synthesis method based on statistical parameters can solve the generalization problem outside the training set, but the synthesis method based on parameters often leads to an over-smoothing effect, making the synthesized singing voice less naturally [25], [26]. The end-to-end synthesis method can learn the mapping relationship between the input music score or lyrics and the acoustic features.

Many methods also prove that the end-to-end singing synthesis based on the deep model can improve the quality of singing voice. Nakamura et al. [27] used CNN for the singing voice synthesis method and the CNN was used to learn the temporal relationship between frame segments of the singing voice. Ren et al. [28] used a forward transformer-based network to perform end-to-end singing voice synthesis, directly generating a linear spectrum and then obtaining the singing voice through the Griffin-Lim vocoder. Shi et al. [29] combined the perceptual entropy loss function with mainstream time sequence models, including RNN, transformer, and conformer for singing voice synthesis. Xue et al. [30] used an acoustic model of the encoder-decoder architecture to perform end-to-end training on frame-level input. In the decoder, the RNN uses the current encoder output and the Mel spectrum of the previous time sequence as input to predict the Mel spectrum of the current time sequence. Lu et al. [31] added F0 prediction and duration prediction for singing voice synthesis based on phoneme modeling based on the FastSpeech model in TTS. Choi et al. [32] used a GAN-based singing synthesis method combined with boundary balance as the objective function and used an autoregressive method to generate a spectrum to solve the time continuity between two adjacent outputs.

*Corresponding author: Jianzong Wang (jzwang@188.com).
time and the composition relationship on the frequency domain.

Our contributions are as follows:
- To alleviate the complexity of the encoder of the music score, we proposed to use the SU-net architecture model in the end-to-end way for the translation between music score embedding to audio spectrum.
- We introduced the stripe pooling module in our proposed method to enhance the learning of the pitch and harmonics information in the spectrum.

II. RELATED WORKS

A. Singing Voice Synthesis

Singing synthesis is a similar task to speech synthesis, with a number of successful applications in pop music and music production. SVS systems generate the singing voice from a musical score. Previous works have conducted studies on SVS from different aspects, including acoustic modeling [34], parametric synthesis [24], [35], lyrics-to-singing alignment [36], and adversarial synthesis [37].

With the successful application of the deep model on many task [38]–[42], many deep model methods were tried on SVS. Kim et al. [24] proposed a Korean SVS system based on a long-short term memory recurrent neural network. Based on the Korean syllable structure, they tried a feature composing method and adopt LSTM for the SVS.

To tackle the difficulty of singing modeling caused by the high sampling rate, Chen et al. [3] proposed HiFiSinger, an SVS system towards high-fidelity singing voice using 48kHz sampling rate. They introduced multi-scale adversarial training in both the acoustic model and vocoder to enhance singing modeling.

Nakamura [22] used convolution neural networks (CNNs) for SVS, through CNN to model the long-term dependencies of singing voices. The learned acoustic feature sequence is generated for each segment that covers long-term frames.

Expect the application of deep models on SVS, there are also some studies that were transferred from TTS. Lu et al. [31] proposed the XiaoiceSing, which follows the main architecture of FastSpeech. While proposing some singing-specific design, it contains features from the musical score (e.g. note pitch and length).

To address the problem of the small amount of training data, Ren et al. [28] developed DeepSinger, which is a multi-lingual multi-singer singing voice synthesis system. It was built from scratch using singing training data mined from music websites.

GANs have been adapted for TTS in recent years. Chandna et al. [25] proposed WGANSing, it used vocoder parameters for acoustic modeling, to separate the influence of pitch and timbre. It facilitates the modeling of the large variability of the pitch in the singing voice.

B. Baseline Method

The CNN based method of singing voice synthesis from [27], named as CNNSing, model the long-term dependencies of singing voices. An acoustic feature sequence is generated for each segment that consists of long-term frames, and a natural trajectory is obtained without the parameter generation algorithm. The acoustic feature was converted to singing voice waveform audio by a WaveNet vocoder.

The U-net based method of singing voice synthesis from [25], named as WGANSing, used the vocoder parameters for acoustic modeling. Wasserstein-GAN model was adapted for singing voice synthesis, the U-net architecture was used in the generator. Finally, the WORLD vocoder was used for acoustic modeling of the singing voice.

III. PROPOSED METHOD

The singing voice synthesis system generally consists of two main modules. Namely, the acoustic model encodes the input musical score and lyrics into acoustic features, and the vocoder generates audio waveform data from the acoustic features. Figure 1 shows the overall architecture of the proposed SU-net based singing voice synthesis system.

A. Musical Score Encoder

Our singing voice synthesis system uses music scores and lyrics phonemes as input. The music score contains the beginning and end of each note. The note’s value is used to determine whether the note is pronounced. Moreover, the onset of all notes is obtained. Lyrics are transformed into 34 phonemes according to the composition of Japanese pronunciation. The labels of the lyrics in the dataset record the corresponding start and end times are in the audio and phonemes units. The representation of phonemes is numerically coded according to the corresponding positions in the complete set of phonemes to obtain all information with the onset and offset.

The frame rate is obtained according to the audio sampling rate and the frame size, and the time stamp sequence of the note and the phoneme are converted to the frame sequence according to the frame rate, and the shortest duration between the music score and the lyrics is used as the aligned embedding length. The encoded note and phoneme are concatenated as the input of the dense layer. Then through two layers of convolution to form the pre-net, the hidden layer output is used as the feature representation of the music score and lyrics.

B. SU-net Architecture Convolution

In this paper, SU-net is proposed to the task of singing voice synthesis. In the SU-net architecture, several convolution
layers with stripe pooling layers and deconvolution layers are stacked. Downsampling in each layer of the convolution layer halves the size and doubles the channel, while the reverse is true in the deconvolution layer. After each convolution layer does a process of stripe pooling layer to learn the stripe information. The low-level high-sampling rate information flows directly to the high-sampling output layer through skip-connections between the convolution layer and the deconvolution layer of the same size.

The two ends of SU-net correspond to the feature representations of music scores and lyrics and the corresponding audio spectrum for singing voice synthesis tasks. To enhance the dependency learning on the spectrum sequence, the previous spectrum is added to the SU-net input. The training phase directly uses the previous segment of the spectrum in the ground truth as input and uses the previous segment of the predicted spectrum in the inference stage. The significant difference between the encoding of music scores and spectrum is not suitable for direct concatenate as SU-net input. Therefore, the input spectrum is first encoded by the convolution layer to a size adapted to the SU-net input. After obtaining the predicted spectrum, the vocoder Griffin-Lim converts it into the corresponding audio.

To effectively capture the long-term dependence of the fundamental frequency of the singing voice and the harmonics in the time dimension, a method of stripe pooling [43] combining horizontal and vertical stripes is append with the harmonics in the vertical axis time dimension so that it can capture long distances in isolated areas relationship. Secondly, maintaining a narrow stripe of pooling kernel shape in the vertical frequency composition dimension is convenient for capturing the relationship between the local fundamental frequency and the harmonic composition and preventing unrelated regional frequency band interference. Integrating this long and narrow pool kernel enables SU-net networks to aggregate global and local frequency band contexts simultaneously. This is essentially different from the traditional pooling of collecting context from a fixed square area.

For the input spectrum or feature map, the horizontal and vertical stripes are pooled, respectively. Mathematically, given the two-dimensional input $x \in \mathbb{R}^{H \times W}$, a spatial extend of pooling $(H, 1)$ or $(1, W)$ is required. The pooling result is subjected to a layer of one-dimensional convolution and expanded in the horizontal and vertical directions. Thus, the output $y^h \in \mathbb{R}^H$ can be calculated as

$$y^h_i = \frac{1}{W} \sum_{0 \leq j < W} x_{i,j}. \quad (1)$$

The output $y^v \in \mathbb{R}^W$ can be written as

$$y^v_j = \frac{1}{H} \sum_{0 \leq i < H} x_{i,j}. \quad (2)$$

$$y_{c,i,j} = y^h_{c,i} + y^v_{c,j}. \quad (3)$$

The results of the horizontal $y^h$ in Equation (1) and vertical $y^v$ in Equation (2) expansion are combined in Equation (3), where $c$ is the channel of the input tensor. Finally, the input frequency spectrum or feature map $x \in \mathbb{R}^{C \times H \times W}$ is dot-multiplied to obtain the output result in Equation(4).

$$z = \text{Scale}(x, \sigma(f(y))), \quad (4)$$

where $f$ is a convolution with kernel of $(1, 1)$ and $\sigma$ is the sigmoid processing, the Scale(.,.) refers to dot-multiply function.

IV. Experiment

A. Dataset

Our experiment uses the public dataset of the Kiritan [44] singing database, which contains 50 Japanese songs sung by the same female singer. The entire dataset’s singing part is about 58 minutes in total, and the audio sampling rate is 96kHz. We resampled the experimental audio data in the data preprocessing and adjusted the sampling rate to 22.05kHz. In the dataset, 45 songs were selected as the training set, and the other five songs were used as the test set.

B. Experiment Settings

In the acoustic model, the embedding layer encodes the note and lyrics phonemes respectively and then concatenates them together. The embedded feature vector of the phoneme is 256, the embedded feature vector of the embedded note is 32, and the final output feature vector of the embedded layer is 288. The previous segment spectrum and embedding features are concatenated as SU-net input after passing through two convolution layers. The convolution layer uses the same Conv1D configuration, the input and output sizes are 513, the kernel size is 5, the stride is 1, and the padding is 2. The SU-net network layer stack structure consists of 7 layers of down-sampling convolution layers, 7 layers of stripe pooling layers, and 7-layers up-sampling deconvolution layers. Each layer uses two-dimensional convolution and two-dimensional deconvolution. The kernel size is 5, the stride is 2, padding is
2, dilation is 1. The stripe pooling module performs horizontal pattern pooling and vertical pattern pooling respectively after each convolution layer, and the pooling adopts averaging.

We constructed two baseline systems of CNN [27] and U-net [25] for comparison to evaluate the performance of our proposed method. To demonstrate the performance of the proposed singing voice synthesis method, we conduct objective evaluations and subjective evaluations shown in section IV-C and section IV-D.

C. Objective Evaluations

In the objective experiment, F0, voiced and unvoiced segments, and spectrum were compared separately. The F0 comparison is depicted in Figure 3. The estimation of F0 uses the pYin [45] algorithm.

It can be seen from Figure 3, the consistency of the overall F0 estimation is relatively high, and the F0 variability of the synthetic song is slight in the local details. When compared with the baseline method of CNN and U-net, the estimated pitch nearly have the same performance. These results can conclude that the three methods can well learn the generation of F0. Usually, singers will add a certain amount of vibrato and tremolo according to note pitch according to music score and personal singing skills to make the song sound more varied. On the other hand, the synthetic song tends to be close to a mean when dealing with changes in those frequencies. From the comparison of the pitch estimation, the proposed method can learn the note information. And the pitch information is controllable by changing the input note series. In the experiment, we further tried to change the key of the notes, and the generated singing voice can precisely rise and fall following the key.

In synthetic songs, the vocal and nonvocal parts can be treated as an evaluation index. As shown in Figure 4, it is a comparison of the consistency of the vocal synthesis according to the music score and the voiced and unvoiced segments in the ground truth of the singing voice in the time domain. The detection of voiced and unvoiced segments is acquired by vocal detection using a mute threshold.

As can be seen from Figure 4, the top subfigure (a) is the ground truth, and the bottom three are the synthetic songs. With vocal detection, the voice frame is marked as 1, and the unvoiced frame is marked as 0. It can be seen that the predicted singing and non-singing of the proposed method based SU-net in the time domain are consistent with the ground truth, and short silent areas are error synthesized nearby frame 800. It could be eliminated by a post-process of temporal smoothing on the vocal detection. While the baseline methods of CNN and U-net have several errors about vocal are synthesized as nonvocal as depicted in the red rectangles. It will lead to a break in the singing voice and affect the continuity of singing.

Furthermore, we inspect the generated spectrogram. The comparison results are shown in Figure 5. Firstly, we can find that all the synthetic songs have lower energy than the ground truth in high-frequency bands. It shows that some frequency details are discarded of the high-frequency band. When compared to the different methods in low-frequency
bands, our proposed method have superior to the baseline methods. As depicted in the red rectangle in the subfigure of Figure 5, the lines of frequency bins from CNN based synthetic are blurred, while the U-net based and SU-net based are more clearly, and the SU-net has most details close to the ground truth.

Furthermore, we used Mel-Cepstral Distortion (MCD) as a total statistic metric on the test dataset. The comparison result of MCD is shown in Table I.

From the comparison result of the MCD metric, the proposed method of SUSing outperforms the baseline methods of CNNSing and WGANSing. The SUSing has a slight improvement over the WGANSing in terms of MCD, and gets an improvement of 1 over the baseline method of CNNSing. Finally, the proposed method achieved a best MCD score of 8.42 on the test song of Song 50. It shows that the add a stripe module to the U-net architecture will improve the learning of the spectrum.

D. Subjective Evaluations

The subjective test scores Mean Opinion Score (MOS) from the sound quality, and naturalness. For the singing voice test, 20 segments of 10 seconds were selected for each model, and 10 subjects were given a score of 1-5. Finally, the scoring results of all subjects in the sample are averaged as the final scoring result.

The MOS results are shown in Table II. From the results, it can be seen that the MOS of the synthesized singing voice achieved 4.12. Compared with the baseline methods of CNNSing [27] and WGANSing [25]. The MOS of the U-net based is higher than the baseline CNN, about 0.07. It shows that the U-net architecture convolution improves the use of the low layer of the so-called high-resolution information. The input of the deconvolution layer not only feed with the adjacent layer output but also get the output with the same shape low layer convolution output.

V. Conclusion

We proposed SU-net for singing voice synthesis named SUSing. The proposed method utilizes the low layer high dimension information flow to high layers of U-net to enhance the learning of temporal and spatial relationships in the spectrum of the singing voice. A stripe pooling module was added to the U-net architecture to learn the stripe information of pitch and harmonics in the spectrum structure. The proposed method was implemented on Kiritan, which is a public dataset of Japanese songs. SU-net was proposed to do the translation from the embedding of music score and lyrics to audio spectrum in an end-to-end way. We showed the proposed method is superior to the baselines of CNN based and the non-stripe U-net based method.

### Table I

| Song | CNNSing [27] | WGANSing [25] | SUSing |
|------|--------------|----------------|--------|
| 46   | 9.61         | 8.73           | 8.37   |
| 47   | 9.29         | 8.61           | 8.38   |
| 48   | 9.82         | 8.85           | 8.52   |
| 49   | 9.91         | 8.92           | 8.42   |
| 50   | 9.36         | 8.67           | 8.42   |

### Table II

| Model       | MOS     |
|-------------|---------|
| Ground Truth | 4.60±0.12 |
| CNNSing [27] | 3.98±0.13 |
| WGANSing [25] | 4.05±0.14 |
| SUSing      | 4.12±0.15 |
References

[1] Y.-H. Yi, Y. Ai, Z.-H. Ling, and L.-R. Dai, “Singing voice synthesis using deep autoregressive neural networks for acoustic modeling,” in 26th Annual Conference of the International Speech Communication Association, 2019.

[2] X. Zhang, Y. Yu, Y. Gao, X. Chen, and W. Li, “Research on singing voice detection based on a long-term recurrent convolutional network with vocal separation and temporal smoothing,” *Electronics*, vol. 9, no. 9, p. 1458, 9 2020.

[3] J. Chen, X. Tan, J. Luan, T. Qin, and T.-Y. Liu, “Hifsinger: Towards high-fidelity neural singing voice synthesis,” *arXiv:2009.01776*, 2020.

[4] Y. Hono, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, “Simsy: A deep neural network-based singing voice synthesis system,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2803–2815, 2021.

[5] X. Qu, J. Wang, and J. Xiao, “Enhancing data-free adversarial distillation with activation regularization and virtual interpolation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3340–3344.

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

[7] Y. Gao, X. Zhang, and W. Li, “Vocal melody extraction via hmrnet-based singing voice separation and encoder-decoder-based F0 estimation,” *Electronics*, vol. 10, no. 3, p. 298, 2021.

[8] J. Kong, J. Kim, and J. Bae, “Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 022–17 033, 2020.

[9] J. Lee, H.-S. Choi, and et al., “Adversarially trained end-to-end korean singing voice synthesis system,” in 20th Annual Conference of the International Speech Communication Association, 2019.

[10] Y. Gu, X. Yin, Y. Rao, Y. Wan, B. Tang, Y. Zhang, J. Chen, Y. Wang, and Z. Ma, “Byteasing: A chinese singing voice synthesis system using duration allocated encoder-decoder acoustic models and wavvern vocoders,” in *12th International Symposium on Chinese Spoken Language Processing*. IEEE, 2021, pp. 1–5.

[11] X. Zhang, J. Qian, Y. Yu, Y. Sun, and W. Li, “Singer identification using deep timbre feature learning with kmn-net,” in 2021 *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2021, pp. 3380–3384.

[12] H. Tang, X. Zhang, J. Wang, N. Cheng, Z. Zeng, E. Xiao, and J. Xiao, “TQAVC: Improving autoencoder voice conversion with text-guided and adversarial training,” in *IEEE Automatic Speech Recognition and Understanding Workshop*. IEEE, 2021, pp. 1–6.

[13] J. Bonada, M. Umbert, and M. Blauuw, “Expressive singing synthesis based on unit selection for the singing synthesis challenge 2016,” in *17th Annual Conference of the International Speech Communication Association*. ISCA, 2016, pp. 1230–1234.

[14] X. Zhang, J. Wang, N. Cheng, and J. Xiao, “Singer identification for metaverse with timbral and middle-level perceptual features,” in *International Joint Conference on Neural Networks*, IJCNN 2022. IEEE, 2022, pp. 1–7.

[15] Y.-R. Chien, H.-M. Wang, and S.-K. Jeng, “Alignment of lyrics with accompanied singing audio based on acoustic-phonetic vowel likelihood modeling,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 1998–2008, 2016.

[16] X. Zhang, J. Wang, N. Cheng, E. Xiao, and J. Xiao, “CycleGEGAN: Cycle generative enhanced adversarial network for voice conversion,” in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU2021)*. IEEE, 2021, pp. 1–6.

[17] C. Gupta, R. Tong, H. Li, and Y. Wang, “Semi-supervised lyrics and solo-singing alignment,” in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, 2018, pp. 600–607.

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

[19] A. Sun, J. Wang, N. Cheng, M. Tantrawenith, Z. Wu, H. Meng, E. Xiao, and J. Xiao, “Reconstructing dual learning for neural voice conversion using relatively few samples,” in *IEEE Automatic Speech Recognition and Understanding Workshop*. IEEE, 2021, pp. 946–953.

[20] Y. Hono, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, “Singing voice synthesis based on generative adversarial networks,” in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2019, pp. 6955–6959.

[21] S. Si, J. Wang, J. Peng, and J. Xiao, “Towards speaker age estimation with label distribution learning,” in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 4618–4622.

[22] K. Nakamura, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, “Singing voice synthesis based on convolutional neural networks,” vol. abs/1904.06868, 2019.

[23] X. Zhang, J. Wang, N. Cheng, and J. Xiao, “Metasid: Singer identification with domain adaptation for metaverse,” in *International Joint Conference on Neural Networks*, IJCNN 2022. IEEE, 2022, pp. 1–7.

[24] J. Kim, H. Choi, J. Park, S. Kim, J. Kim, and M. Hahn, “Korean singing voice synthesis system based on an lstm recurrent neural network,” in *17th Annual Conference of the International Speech Communication Association*, 2018, pp. 1551–1555.

[25] P. Chanda, M. Blauuw, J. Bonada, and E. Gómez, “Wgansing: A multi-voice singing voice synthesizer based on the wasserstein-gan,” in *27th European Signal Processing Conference*. IEEE, 2019, pp. 1–5.

[26] X. Zhang, J. Wang, N. Cheng, and J. Xiao, “Mcdn-sid: Multi-scale dilated convolution network for singer identification,” in *International Joint Conference on Neural Networks*, IJCNN 2022. IEEE, 2022, pp. 1–7.

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

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

[29] J. Shi, S. Guo, N. Hsu, Y. Zhang, and Q. Jin, “Sequence-to-sequence singing voice synthesis with perceptual entropy loss,” in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2021, pp. 76–80.

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

[31] P. Lu, J. Wu, J. Luan, X. Tan, and L. Zhou, “Xiaoicesing: A high-quality and integrated voice singing voice synthesizer,” in *Proceedings of the 18th International Conference of the International Speech Communication Association*, ISCA, 2020, pp. 1306–1310.

[32] S. Choi, W. Kim, S. Park, S. Yong, and J. Nam, “Korean singing voice synthesis based on auto-regressive boundary equilibrium gan,” in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2020, pp. 7234–7238.

[33] A. Jansson, E. J. Humphrey, N. Montecchio, R. M. Bittner, A. Kumar, and T. Weyde, “Singing voice separation with deep u-net convolutional networks,” in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, 2017, pp. 745–751.

[34] M. Nishimura, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, “Singing voice synthesis based on deep neural networks,” in *17th Annual Conference of the International Speech Communication Association*, 2016.

[35] X. Zhang, J. Wang, N. Cheng, and J. Xiao, “Tdass: Target domain adaptation speech synthesis framework for multi-speaker low-resource tts,” in *International Joint Conference on Neural Networks*, IJCNN 2022. IEEE, 2022, pp. 1–7.

[36] Y.-R. Chien, H.-M. Wang, and S.-K. Jeng, “Alignment of lyrics with accompanied singing audio based on acoustic-phonetic vowel likelihood modeling,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 1998–2008, 2016.
[37] Y. Hono, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, “Singing
voice synthesis based on generative adversarial networks,” in 2019 IEEE
International Conference on Acoustics, Speech and Signal Processing,
2019, pp. 6955–6959.

[38] Y. Sun, X. Zhang, X. Chen, Y. Yu, and W. Li, “Investigation of singing
voice separation for singing voice detection in polyphonic music,” in
Proceedings of the 9th Conference on Sound and Music Technology,
2021, pp. 1–6.

[39] B. Zhao, X. Zhang, J. Wang, N. Cheng, and J. Xiao, “anspeech: Speaker-
guided conditional variational autoencoder for zero-shot multi-speaker
text-to-speech,” in 2022 IEEE International Conference on Acoustics,
Speech and Signal Processing. IEEE, 2022, pp. 1–5.

[40] Q. Wang, X. Zhang, J. Wang, N. Cheng, and J. Xiao, “Drvc: A frame-
work of any-to-any voice conversion with self-supervised learning,” in
2022 IEEE International Conference on Acoustics, Speech and Signal
Processing. IEEE, 2022, pp. 1–5.

[41] H. Tang, X. Zhang, J. Wang, N. Cheng, and J. Xiao, “Avqvc: One-
shot voice conversion by vector quantization with applying contrastive
learning,” in 2022 IEEE International Conference on Acoustics, Speech
and Signal Processing. IEEE, 2022, pp. 1–5.

[42] L. Wang, H. Zhu, X. Zhang, S. Li, and W. Li, “Transfer learning
for music classification and regression tasks using artist tags,” in
Proceedings of the 7th Conference on Sound and Music Technology
(CSMT2019). Springer, 2020, pp. 81–89.

[43] Q. Hou, L. Zhang, M. Cheng, and J. Feng, “Strip pooling: Rethinking
spatial pooling for scene parsing,” in 2020 IEEE/CVF Conference on
Computer Vision and Pattern Recognition. IEEE, 2020, pp. 4002–4011.

[44] I. Ogawa and M. Morise, “Tohoku kiritan singing database: A singing
database for statistical parametric singing synthesis using japanese pop
songs,” Acoustical Science and Technology, vol. 42, no. 3, pp. 140–145,
2021.

[45] M. Mauch and S. Dixon, “Pyin: A fundamental frequency estimator
using probabilistic threshold distributions,” in 2014 IEEE International
Conference on Acoustics, Speech and Signal Processing. IEEE, 2014,
pp. 659–663.