FASTSVC: FAST CROSS-DOMAIN SINGING VOICE CONVERSION WITH FEATURE-WISE LINEAR MODULATION

Songxiang Liu[1,*], *Yuewen Cao*[1,*], *Na Hu*[2], *Dan Su*[2], *Helen Meng*[1]

[1]Human-Computer Communications Laboratory, The Chinese University of Hong Kong
[2]Tencent AI Lab

ABSTRACT

This paper presents FastSVC, a light-weight cross-domain singing voice conversion (SVC) system, which can achieve high conversion performance, with inference speed 4x faster than real-time on CPUs. FastSVC uses Conformer-based phoneme recognizer to extract singer-agnostic linguistic features from singing signals. A feature-wise linear modulation based generator is used to synthesize waveform directly from linguistic features, leveraging information from sine-excitation signals and loudness features. The waveform generator can be trained conveniently using a multi-resolution spectral loss and an adversarial loss. Experimental results show that the proposed FastSVC system, compared with a computationally heavy baseline system, can achieve comparable conversion performance in some scenarios and significantly better conversion performance in other scenarios. Moreover, the proposed FastSVC system achieves desirable cross-lingual singing conversion performance. The inference speed of the FastSVC system is 3x and 70x faster than the baseline system on GPUs and CPUs, respectively.

Index Terms— Singing voice conversion, cross-domain, generative adversarial network

1. INTRODUCTION

Human singing is an important way of information transmission, emotional expression and entertainment. Enabling machine the ability to produce high-fidelity singing voice can enrich the way of human-computer interaction. This paper focuses on a singing synthesis related task, i.e., singing voice conversion (SVC), which aims at converting the voice of one singer to that of other singers without changing the underlying content and melody.

In terms of whether parallel singing datasets are used during training, which are composed of paired samples among singers singing the same content, current SVC approaches can be categorized into two classes: parallel SVC and non-parallel SVC. Most initial attempts for SVC belong to the parallel SVC class, which model parallel training samples using statistical methods, such as Gaussian mixture model (GMM) based many-to-many eigenvoice conversion [1], direct waveform modification based on spectrum difference [2, 3]. Artificial neural network (ANN) based approaches are also proposed to improve conversion performance [4, 5].

Since parallel singing datasets are expensive to collect in large scale, many non-parallel SVC approaches are proposed. WaveNet [6] autoencoder based unsupervised SVC model is trained to convert among singers appeared in the training set [7], where an adversarial speaker classifier is incorporated to disentangle singer information from the encoder output. To further improve this method, PitchNet [8] adopts an additional domain fusion term on the pitch to remove pitch information from the encoder output. Variational autoencoder (VAE) [9], generative adversarial network (GAN) [10], and phonetic posteriorgram (PPG) [11] based approaches are also investigated for non-parallel SVC. However, these methods either use acoustic features from conventional vocoders (e.g., WORLD [12]) or mel spectrograms as intermediate representations during conversion, which may bound the audio quality.

A very recent unsupervised cross-domain SVC approach (UCD-SVC) [13], which combines a linguistic extractor with a WaveNet based waveform generator, can convert any source singer to a target speaker/singer appeared in the training set (referred to as any-to-many SVC). UCD-SVC uses a pure convolution network based non-autoregressive model for the waveform-based generator, resulting in very low latency during inference on GPUs. The usage of pitch perceptual loss and automatic speech recognition (ASR) perceptual loss effectively boost the conversion performance. Moreover, UCD-SVC can conduct cross-domain training, i.e., the model can be trained using either speech or singing datasets. However, during conversion UCD-SVC uses three computationally heavy neural networks in the pipeline: the CREPE model [14] for fundamental frequency (F0) computation, Jasper based wave-to-letter acoustic model [15] for linguistic feature extraction, and the WaveNet based waveform generator for audio synthesis. This makes the UCD-SVC system have many parameters, which hinders it from conducting singing voice conversion on CPUs efficiently. Moreover, the training process is complicated and slow.

This paper presents FastSVC, a light-weight cross-

*Work done during internship at Tencent AI Lab.

978-1-6654-3864-3/21/$31.00 ©2021 IEEE

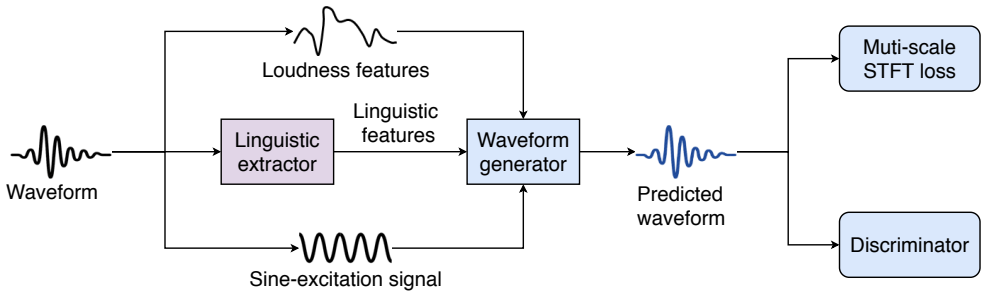

Fig. 1. Schematic diagram of the proposed FastSVC system.

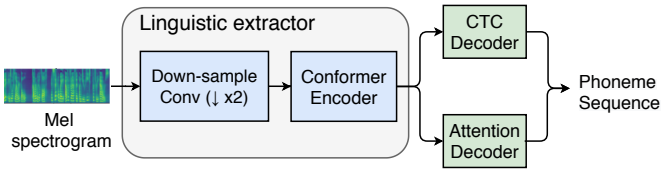

Fig. 2. Hybrid CTC-attention Conformer based model for phoneme recognition.

domain SVC system, which can achieve high conversion performance in terms of audio fidelity and voice similarity from any source singer, with inference speed 4x faster than real-time on CPUs. The proposed FastSVC approach holds all the merits of UCD-SVC: 1) can conduct cross-domain training; 2) can achieve any-to-many SVC; 3) has desirable singing voice conversion performance. FastSVC takes advantage of recent progress in light-weight end-to-end ASR acoustic modeling, information fusion in deep neural networks and GAN based waveform generative modeling. Specifically, FastSVC uses Conformer [16] based phoneme recognizer to extract singer-agnostic linguistic features from singing signals. A feature-wise linear modulation (FiLM) [17] based generator is used to synthesize waveform directly from linguistic features, effectively fusing information from sine-excitation signals and loudness features. The waveform generator can be trained using only a combination of a multi-scale spectral loss and an adversarial loss, which is simpler and faster than the UCD-SVC.

The rest of this paper is organized as follows: Section 2 presents the proposed FastSVC system. Experiments are described in Section 3 and Section 4 concludes this paper.

2. PROPSOSED METHOD

The proposed FastSVC system concatenates a linguistic extractor with a waveform generator, as illustrated in Fig. 1. The linguistic extractor is used to compute singer/speaker-agnostic linguistic features from singing/speech signals, while the waveform generator directly outputs raw waveform from linguistic features, conditioned on sine-excitation signals and loudness features. The linguistic extractor and the waveform generator are trained sequentially, since the waveform generator training process requires linguistic features extracted from a well-trained linguistic extractor. Details of the linguistic extractor and waveform generator are presented in Section 2.1 and Section 2.2, respectively.

2.1. Linguistic extractor

The UCD-SVC system adopts a pretrained Jasper ASR acoustic model [15], to extract singer/speaker-agnostic linguistic features from wave signals. However, the Jasper model is computationally heavy with more than 300 million parameters. One goal of this study is to make the whole SVC system parameter efficient and light-weight, such that inference on modern CPUs can be feasible and fast. The very recent Conformer based ASR acoustic model achieves state-of-the-art recognition performance on LibriSpeech corpus [18]. It is well known that the Transformer models [19] are good at capturing content-based global interactions, while convolution neural network (CNN) models exploit local features effectively. The Conformer model combines the merits of both the Transformer models and CNN models and also make itself parameter efficient.

Attracted by the advantages of the Conformer model, this study adopts a Conformer based network structure for the linguistic extractor, as illustrated in Fig. 2. Specifically, we obtain a linguistic extractor from an end-to-end hybrid CTC-attention phoneme recognizer, where the Conformer encoder follows the structure of the small version presented in Table 1 of [16]. The input spectral features are 80-dimensional log mel spectrograms, on which utterance-level mean-variance normalization is conducted before feeding into the recognizer model. The down-sample layer adopts a strided 2D convolutional structure to down-sample the input mel spetrograms in time scale by a factor of 2, where kernel-size is 4, stride is 2 and output channels is 160. Then the hidden feature maps from the down-sample layer are fed into the Conformer encoder. Then a CTC decoder and attention decoder take the encoder output as input to predict phoneme sequences. The CTC decoder contains one fully-connected (FC) layer. The attention decoder uses location-sensitive attention [20] mech-

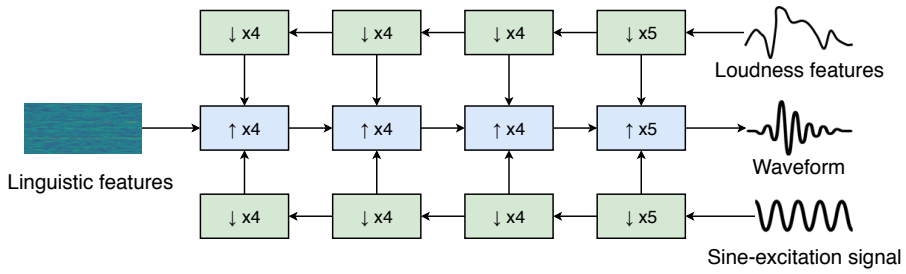

Fig. 3. Schematic diagram of the waveform generator in the proposed FastSVC system.

anism and has one decoder LSTM layer with hidden size of 320. Denote mel spectrogram features as X and phoneme sequence as Y, the training loss of the phoneme recognizer is a linear combination of the CTC and attention losses:

$$\mathcal{L} = \lambda \mathcal{L}_{ctc}(Y|X) + (1-\lambda)\mathcal{L}_{att}(Y|X) \tag{1}$$

where $\lambda \in [0, 1]$ is a hyper-parameter weighting the CTC objective $\mathcal{L}_{ctc}$ and the attention objective $\mathcal{L}_{att}$. In this paper, we set λ to be 0.5.

The phoneme recognizer is trained with the LibriSpeech corpus (960 hours). After training, we drop the CTC module and attention decoder from the phoneme recognizer and use the remaining part as the linguistic extractor, which only contains 9 million parameters.

2.2. Waveform generator

As shown in Fig. 1, the waveform generator takes linguistic features, sine-excitation signals and loudness features as input, and synthesizes waveform directly. Linguistic features provide important pronunciation traits for singing voice generation, while sine-excitation signals are melody presentations which proved to be better than F0 contours for SVC tasks [13]. Loudness features make better energy rendering possible in the generated waveform. An overview of the novel generator model structure is illustrated in Fig. 3, where the blocks with ↑ and ↓ means up-sample blocks and down-sample blocks respectively. The intuitions behind this using two U-shape branches is to fully fuse information from the sine-excitation signals and loudness features into the waveform generation process in different time scales.

2.2.1. Sine-excitation signals and loudness features

Sine-excitation signals are computed from the F0 values. Following the NSF models [21], F0 values in frame-rate are first upsampled by linear interpolation to audio-rate and then are regarded as instantaneous frequencies. In voiced segments, the excitation signal are presented as sine waveform, while in unvoiced regions, the excitation signal is represented by Gaussian noise. Denote audio-rate F0 sequence as $\boldsymbol{f}_{1:T}$, following [21], a sine-excitation signal $\boldsymbol{e}_{1:T}$ is computed as:

$$\boldsymbol{e}_t = \begin{cases} 0.1 \sin\left(\sum_{k=1}^{t} 2\pi \frac{f_k}{f_s} + \phi\right) + n_t & \text{if } f_t > 0 \\ 100 n_t & \text{if } f_t = 0 \end{cases} \tag{2}$$

where $n_t \sim \mathcal{N}(0, 0.003^2)$, $\phi \in [-\pi, \pi]$ is a random initial phase and f_s is the waveform sampling rate.

A-weighting mechanism of the power spectrum, which puts greater emphasis on higher frequencies, is adopted to compute loudness features in this paper. The computation process is identical to that as shown in [22]. This paper uses a hop-size of 64 when computing loudness features, which are up-sampled to audio-rate using linear interpolation operation before being fed into the generator.

2.2.2. Generator model details

The up-sample blocks and down-sample blocks as shown in Fig. 3 can adopt arbitrary convolutional network structure. The model architecture of the building blocks in the generator is adapted from [23] and proper modifications are made for the SVC tasks. Details of the network structure of building blocks in the waveform generator is illustrated in Fig. 4. To convert linguistic features with hop-size of 320 samples into waveform, four up-sample blocks are applied to gradually up-sample the temporal dimension by factors of 4, 4, 4, 5 with the number of channels of 192, 96, 48, 24 respectively. The dilation rates are 1, 3, 9, 27 in all up-sample blocks. Down-sample blocks downsample the time dimension of the audio-rate sine-excitation signals and loudness features. The number of channels of the down-sample blocks matches the number of up-sample blocks correspondingly. The dilation rates are 1, 2, 4 in all down-sample blocks. LeakyReLU activation function uses a negative slope of 0.2.

The feature-wise linear modulation (FiLM) [17] module is used to fuse information from sine-excitation signals and loudness features with the linguistic features, which produces scale and shift vectors given inputs as shown in Fig.4(c). FiLM modules have the same number of convolution channels as their corresponding up-sample blocks. The feature-wise affine operation as shown in Fig.4(a) is conducted as

$$(\gamma_{sine} + \gamma_{loudness}) \odot U_{linguistic} + \xi_{sine} + \xi_{loudness}, \tag{3}$$

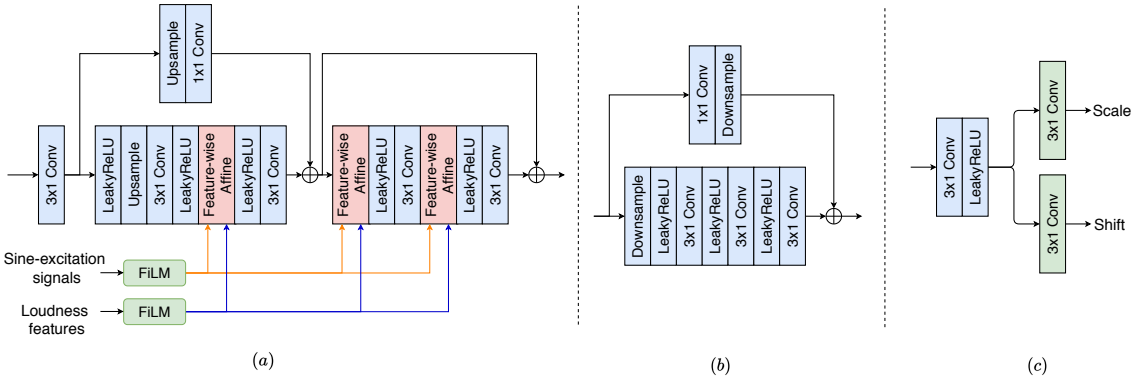

Fig. 4. Network details of building blocks of the waveform generator in the proposed FastSVC system. (a) Network details of the up-sample block. (b) Network details of the down-sample block. (c) The FiLM block appeared in (a).

where γ's and ξ's represent the scale and shift vectors from the FiLM modules, $U_{linguistic}$ is the up-sampled linguistic features and $\odot$ denotes the Hadamard product.

In multi-speaker/singer SVC models, the waveform generator has an additional speaker/singer embedding table. The speaker/singer identity information is fused into the up-sample blocks by adding the results from each feature-wise affine operation with speaker/singer embedding vectors. To remove speaker/singer identity information from the results of each feature-wise affine operation, instance normalization [24] without affine transformation is performed before combination with speaker/singer embedding vectors.

2.2.3. Training objectives

Similar to [13], the waveform generator is trained under the least-squares GAN [25] setup, with combination of a multi-scale STFT loss [26]. The discriminator module as shown in Fig. 3 adopts the same multi-scale discriminator architecture presented in MelGAN [27]. In this section, we denote the three sub-discriminators as $D_k, \forall k \in [1, 2, 3]$, the groud-truth waveform as x and the reconstructed waveform as $\hat{x}$. The generator's adversarial loss $\mathcal{L}_{adv}$ is:

$$\mathcal{L}_{adv} = \frac{1}{k}\sum_{k} ||1 - D_k(\hat{x})||_2. \tag{4}$$

The discriminator loss $\mathcal{L}_D$ is computed as:

$$\mathcal{L}_D = \frac{1}{k}\sum_{k} (||1 - D_k(x)||_2 + ||D_k(\hat{x})||_2). \tag{5}$$

The multi-scale STFT loss is computed as:

$$\mathcal{L}_{stft} = \frac{1}{|M|}\sum_{m \in M} (\frac{||S_m - \hat{S}_m||_2}{||S_m||_2} + \frac{||\log S_m - \log \hat{S}_m||_1}{N}), \tag{6}$$

where S_m and $\hat{S}_m$ are the STFT magnitudes computed from x and $\hat{x}$ respectively, with FFT sizes of $m \in M = [2048, 1024, 512, 256, 128, 64]$ and with 75% overlap. N is the number of elements.

The final waveform generator loss $\mathcal{L}_G$ is a linear combination of the adversarial loss and the multi-scale STFT loss as:

$$\mathcal{L}_G = \mathcal{L}_{stft} + \alpha\mathcal{L}_{adv}, \tag{7}$$

where in this paper, $\alpha = 2.5$.

3. EXPERIMENTS

3.1. Experimental setup

We choose the UCD-SVC system as the baseline. Both cross-domain and in-domain SVC performance are compared between the UCD-SVC system and the proposed FastSVC system. We also report their cross-lingual SVC performance [28].

Three open-source English datasets are used, which are the LJ-Speech corpus [29], the VCTK corpus [30] and the NUS-48E corpus [31]. All audio is resampled to 16kHz with mono channel. Datasets are randomly split into train-validation-test sets according to a 90%-5%-5% partition. We compare the any-to-one cross-domain (**A2O-CD**) SVC performance of the UCD-SVC and FastSVC systems by training the models with the single-speaker LJ-Speech corpus, and their any-to-many cross-domain (**A2M-CD**) SVC performance by training the models with the multi-speaker VCTK corpus (108 speakers in total). Any-to-many in-domain (**A2M-ID**) and cross-lingual (**CL**) SVC performance of the UCD-SVC and FastSVC systems are examined by training the models with the multi-singer NUS-48E corpus (12 singers in total). During conversion, source signals are chosen from the NUS-48E test set, except that we use internal Chinese source samples when conducting cross-lingual SVC since there is no open-source Chinese singing dataset.

All models in the UCD-SVC and FastSVC systems are trained at least 600k steps until their losses converge, with

Table 1. Mean opinion score (MOS) results.

Scenario	Naturalness		Similarity	
	UCD-SVC	FastSVC	UCD-SVC	FastSVC
A2O-CD	3.94±0.09	4.00±0.08	3.35±0.23	3.56±0.20
A2M-CD	3.06±0.08	3.52±0.10	2.67±0.18	3.27±0.22
A2M-ID	3.47±0.07	3.48±0.06	3.17±0.15	3.58±0.19
CL	2.92±0.07	3.09±0.08	3.15±0.19	3.26±0.19
Recording	4.62±0.16		-	

Table 2. Voice similarity.

Scenario	Source-target	Converted-target	
		UCD-SVC	FastSVC
A2O-CD	0.082	0.555	**0.676**
A2M-CD	0.124	**0.588**	0.433
A2M-ID	0.234	0.712	**0.785**
CL	0.274	**0.821**	0.801

batches of 32 one-second long audio segments. The ADAM optimizer [32] with a learning rate of 0.001 is used, where learning rate decays by 0.5 every 100k steps. The discriminator joins the training process after 100k steps. F0 values are extracted using the WORLD vocoder [12] in the FastSVC system.

3.2. Subjective evaluation

Subjective evaluation in terms of both the naturalness and voice similarity of the converted singing samples are conducted[1]. The standard 5-scale mean opinion score (MOS) test is adopted. In the MOS tests for evaluating naturalness, each group of stimuli contains recording samples which are randomly shuffled with the samples converted by the UCD-SVC and FastSVC systems before presented to raters. In MOS voice similarity tests, converted samples are directly compared with the target singers' reference recordings. At least 24 samples are rated for the compared systems in each conversion scenario. We invite 20 Chinese speakers who are also proficient in English to participate in the MOS tests.

The subjective results are presented in Table 1. We can see that in all SVC scenarios, the FastSVC achieves better voice similarity. In terms of audio naturalness, the FastSVC achieves comparable conversion performance to the UCD-SVC system in the any-to-one cross-domain (A2O-CD) and any-to-many in-domain (A2M-ID) scenarios. In the any-to-many cross-domain (A2M-CD) and cross-lingual (CL) scenarios, the FastSVC achieves significantly better performance than the UCD-SVC. The subjective results verify the efficacy of the network structure design and training loss selection in the FastSVC system.

3.3. Objective evaluation

While MOS is a desired measure for audio naturalness/fidelity in singing voice synthesis tasks, voice similarity is more difficult to subjectively measured since human perception of voice similarity can vary when a singer/speaker utters the same content with different pitch patterns. This can be reflected in Table 1, where the standard variances of voice similarity MOS values are much bigger than those of naturalness MOS values.

Therefore, we adopt a pre-trained end-to-end speaker recognition model named RawNet2 [33] to objectively measure the voice similarity of the converted singing samples. We measure cosine similarities between embedding vectors of audio samples and the desired target speaker embedding vectors before and after conversion, where all embedding vectors are computed by the RawNet2 and singer/speaker embedding vectors are obtained by averaging his/her training audio samples. The voice similarity results are illustrated in Table 2. We can see that both the UCD-SVC and the FastSVC systems can significantly improve cosine similarity of audio sample to a desired target singer/speaker after conversion. It is worthy to note that these objective results are not consistent with the subjective results in Table 1, one possible reason is that the pre-trained RawNet2 model is trained using only speech data. It should be better to train a RawNet2 model for speaker/singer embedding vector computation; but we can not access to a large multi-singer corpus during the submission of this paper, this is to be solved in the future work.

Table 3. Inference speed comparison between the UCD-SVC and FastSVC systems. Pytorch implementation without hardware optimization for an Nvidia Tesla P40 GPU and Intel(R) Xeon(R) E5-2680(v4) CPU @ 2.40GHz.

System	Parameters	RTF (GPU)	RTF (CPU)
UCD-SVC	368.4M	0.103	17.5
FastSVC (Ours)	**11.9**M	**0.031**	**0.248**

3.4. Inference speed

The inference speed benchmark results of the compared UCD-SVC and FastSVC systems on both GPUs and CPUs are presented in Table 3. All models are implemented with the Pytorch toolkit without any hardware optimization. The proposed FastSVC system has much less number of parameters (11.9M) than the UCD-SVC system (368.4M). Inference speed of the FastSVC system is 3x faster on GPUs and 70x faster on CPUs than the UCD-SVC system. The proposed FastSVC system achieves a real-time factor (RTF) of 0.248 (i.e., 4x faster than real-time) on modern CPUs.

4. CONCLUSIONS

In this paper, we have presented FastSVC, a parameter efficient and light-weight cross-domain SVC system, which can achieve superior conversion performance in terms of audio naturalness and voice similarity. The inference speed of the FastSVC system is very fast in both GPUs and CPUs (with real-time factors (RTFs) of 0.031 and 0.248, respectively),

[1]Audio demo: `https://nobody996.github.io/FastSVC/`.

which means that FastSVC can be deployed for low-latency real-world applications. Future works include further reducing the parameter size of the FastSVC system and investigating its singer adaptation behavior in the low-resource scenario.

5. ACKNOWLEDGEMENTS

This project is partially supported by a grant from the HKSAR Government's Research Grants Council General Research Fund (Project no. 14208817).

6. REFERENCES

[1] T. Toda, Y. Ohtani, and K. Shikano, "One-to-many and many-to-one voice conversion based on eigenvoices," in *ICASSP*. IEEE, 2007, vol. 4, pp. IV–1249.

[2] K. Kobayashi, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, "Statistical singing voice conversion with direct waveform modification based on the spectrum differential," in *INTERSPEECH*, 2014.

[3] K. Kobayashi, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, "Statistical singing voice conversion based on direct waveform modification with global variance," in *INTERSPEECH*, 2015.

[4] Y. Hono, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Singing voice synthesis based on generative adversarial networks," in *ICASSP*. IEEE, 2019, pp. 6955–6959.

[5] B. Sisman, K. Vijayan, M. Dong, and H. Li, "Singan: Singing voice conversion with generative adversarial networks," in *APSIPA ASC)*. IEEE, 2019, pp. 112–118.

[6] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *ISCA SSW*, 2016.

[7] E. Nachmani and L. Wolf, "Unsupervised singing voice conversion," *INTERSPEECH*, 2019.

[8] C. Deng, C. Yu, H. Lu, C. Weng, and D. Yu, "Pitchnet: Unsupervised singing voice conversion with pitch adversarial network," in *ICASSP*. IEEE, 2020, pp. 7749–7753.

[9] Y.-J. Luo, C.-C. Hsu, K. Agres, and D. Herremans, "Singing voice conversion with disentangled representations of singer and vocal technique using variational autoencoders," in *ICASSP*. IEEE, 2020, pp. 3277–3281.

[10] J. Lu, K. Zhou, B. Sisman, and H. Li, "Vaw-gan for singing voice conversion with non-parallel training data," *arXiv preprint arXiv:2008.03992*, 2020.

[11] Z. Li, B. Tang, X. Yin, Y. Wan, L. Xu, C. Shen, and Z. Ma, "Ppg-based singing voice conversion with adversarial representation learning," *arXiv preprint arXiv:2010.14804*, 2020.

[12] M. Morise, F. Yokomori, and K. Ozawa, "World: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[13] A. Polyak, L. Wolf, Y. Adi, and Y. Taigman, "Unsupervised cross-domain singing voice conversion," *INTERSPEECH*, 2020.

[14] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," in *ICASSP*. IEEE, 2018, pp. 161–165.

[15] J. Li, V. Lavrukhin, B. Ginsburg, R. Leary, O. Kuchaiev, J. M. Cohen, H. Nguyen, and R. T. Gadde, "Jasper: An end-to-end convolutional neural acoustic model," *INTERSPEECH*, 2019.

[16] A. Gulati et al., "Conformer: Convolution-augmented transformer for speech recognition," *INTERSPEECH*, 2020.

[17] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," *arXiv preprint arXiv:1709.07871*, 2017.

[18] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*. IEEE, 2015, pp. 5206–5210.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017, pp. 5998–6008.

[20] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *NeurIPS*, 2015, pp. 577–585.

[21] X. Wang, S. Takaki, and J. Yamagishi, "Neural source-filter-based waveform model for statistical parametric speech synthesis," in *ICASSP*. IEEE, 2019, pp. 5916–5920.

[22] L. Hantrakul, J. H. Engel, A. Roberts, and C. Gu, "Fast and flexible neural audio synthesis.," in *ISMIR*, 2019, pp. 524–530.

[23] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "Wavegrad: Estimating gradients for waveform generation," *arXiv preprint arXiv:2009.00713*, 2020.

[24] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[25] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *ICCV*, 2017, pp. 2794–2802.

[26] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *ICASSP*. IEEE, 2020, pp. 6199–6203.

[27] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," in *NeurIPS*, 2019, pp. 14910–14921.

[28] L. Sun, H. Wang, S. Kang, K. Li, and H. M. Meng, "Personalized, cross-lingual tts using phonetic posteriorgrams.," in *INTERSPEECH*, 2016, pp. 322–326.

[29] K. Ito and L. Johnson, "The lj speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[30] J. Yamagishi et al., "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," 2019.

[31] Z. Duan, H. Fang, B. Li, K. C. Sim, and Y. Wang, "The nus sung and spoken lyrics corpus: A quantitative comparison of singing and speech," in *APSIPA ASC*. IEEE, 2013, pp. 1–9.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ICLR*, 2015.

[33] J.-w. Jung, S.-b. Kim, H.-j. Shim, J.-h. Kim, and H.-J. Yu, "Improved rawnet with feature map scaling for text-independent speaker verification using raw waveforms," *INTERSPEECH*, pp. 3583–3587, 2020.