INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

ARVC: An Auto-Regressive Voice Conversion System Without Parallel Training Data

Zheng Lian[1,2]*, Zhengqi Wen*[1]*, Xinyong Zhou*[3]*, Songbai Pu*[4]*, Shengkai Zhang*[4] *and Jianhua Tao*[1,2,5]

[1]National Laboratory of Pattern Recognition, CASIA, Beijing
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing
[3]School of Computer Science, Northwestern Polytechnical University, Xi'an
[4]MOMO DLLAB, Beijing
[5]CAS Center for Excellence in Brain Science and Intelligence Technology, Beijing

{zheng.lian, zqwen}@nlpr.ia.ac.cn

Abstract

Voice conversion (VC) is to convert the source speaker's voice to sound like that of the target speaker without changing the linguistic content. Recent work shows that phonetic posteriorgrams (PPGs) based VC frameworks have achieved promising results in speaker similarity and speech quality. However, in practice, we find that the trajectory of some generated waveforms is not smooth, thus causing some voice error problems and degrading the sound quality of the converted speech. In this paper, we propose to advance the existing PPGs based voice conversion methods to achieve better performance. Specifically, we propose a new auto-regressive model for any-to-one VC, called Auto-Regressive Voice Conversion (ARVC). Compared with conventional PPGs based VC, ARVC takes previous step acoustic features as the inputs to produce the next step outputs via the auto-regressive structure. Experimental results on the CMU-ARCTIC dataset show that our method can improve the speech quality and speaker similarity of the converted speech.

Index Terms: Auto-regressive voice conversion (ARVC), Phonetic posteriorgrams (PPGs), LPCNet

1. Introduction

Voice conversion (VC) aims to modify the source speaker's voice to sound like that of the target speaker while keeping the linguistic content unchanged. VC is an important research topic due to its wide applications, such as development of personalized speaking aids for speech-impaired subjects [1], novel vocal effects of singing voices [2, 3], and a voice changer to generate various types of expressive speech [4].

The conventional voice conversion approach usually needs parallel training data, which contains pairs of the same transcription utterances spoken by different speakers. Parallel VC first aligns acoustic units between source and target speech by dynamic time warping, then a conversion model is learned to map speech from source to target speaker. Several statistical conversion models have been proposed, including Gaussian mixture models (GMM) [5], exemplar-based models [6] and neural networks [7]. Recently, the sequence-to-sequence (seq2seq) model with attention has been studied for parallel VC [8, 9]. Compared with conventional methods, this method can achieve better naturalness and speaker similarity. However, problems such as mispronunciation and training instability have also been observed while training the seq2seq VC model [9].

When parallel training data is unavailable, there are also some methods for non-parallel VC. Variational autoencoder (VAE) [10] has been successfully proposed for non-parallel VC [11, 12]. However, VAE suffers from over-smoothing. To address this problem, generative adversarial network (GAN) [13] and its variants (such as CycleGAN [14, 15] and StarGAN [16, 17]) use a discriminator that amplifies this artifact in the loss function. However, these methods are hard to train, and the discriminator's discernment may not correspond well to human auditory perception, thus degrading the sound quality of the converted speech. Recently, there is another track of research [18, 19] that applies phonetic posteriorgrams (PPGs) for non-parallel VC. PPGs are of frame-level linguistic information representations obtained from the speaker-independent automatic speech recognition (SI-ASR) system. The PPGs based VC frameworks mainly have two key components: the conversion model and the vocoder. The conversion model converts PPGs extracted from the source speech into acoustic features of the target speaker. Then the vocoder uses these converted features to synthesize the speech waveform of the target speaker. However, in practice, we find that the trajectory of some generated waveforms is not smooth, thus causing some voice errors.

Vocoders influence the speech quality of the converted speech. Several parametric vocoders have been proposed for VC, including STRAIGHT [20] and WORLD [21]. However, these vocoders limit the quality of generated speech. To deal with this problem, neural vocoders are widely studied and utilized for speech generation. WaveNet [22] is one of the most successful neural vocoders, which is proposed for direct waveform modeling and generation in a data-driven manner. However, since WaveNet relies on sequential generation of one audio sample at a time, it is hard to deploy in a real-time production setting. Recently, an efficient neural vocoder, called LPCNet [23] is proposed. Compared with WaveNet, LPCNet can generate speech in real time. Meanwhile, since LPCNet depends directly on the linear predictive coding filter shape, it can better control over the outputs of the spectral shape. Therefore, we apply LPCNet vocoder for speech generation in this paper.

In this paper, we propose a new auto-regressive model for any-to-one voice conversion, called Auto-Regressive Voice Conversion (ARVC). As shown in Figure 1, ARVC produces the next step acoustic features based on two inputs: 1) the predicted acoustic features of the previous step; 2) the PPGs extracted from the source speech. Our ARVC is an extension of conventional seq2seq base VC [8, 9] and PPGs based VC [18, 19]. Compared with conventional seq2seq base VC [8, 9], ARVC removes the attention-based duration conversion module since PPGs already contain the duration information, thus reducing mispronunciation and improving training stability. Compared with conventional PPGs based VC [18, 19], ARVC takes pre-

http://dx.doi.org/10.21437/Interspeech.2020-1715

vious step acoustic features as the inputs to produce the next step outputs via the auto-regressive structure, thus generating smooth trajectory and causing less voice error problems.

The main contributions of this paper include three aspects: 1) We propose a novel framework, ARVC, for any-to-one voice conversion. Our ARVC is the extension of conventional approaches, which incorporates previous step outputs via the auto-regressive structure; 2) We apply an efficient neural vocoder LPCNet for speech synthesis, since LPCNet can better control over the outputs of the spectral shape, and is able to generate speech in real time [23]; 3) Experimental results on the popular benchmark datasets CMU-ARCTIC demonstrate the effectiveness of our ARVC. Our method provides a significant increase in the speech quality and speaker similarity.

2. Proposed Method

A block diagram of our auto-regressive model for voice conversion is shown in Figure 1. It consists of three key components: (1) Encoder; (2) Decoder; (3) Waveform synthesis. Specifically, we use the frame-level linguistic features, PPGs, as the inputs. The encoder maps the input PPGs into the context-dependent representations. Then the decoder predicts acoustic features from the encoder outputs. In the end, LPCNet is conditioned on the predicted acoustic features for waveform synthesis.

During training, the input speech is the same as the output speech. These waveforms are selected from the target speaker's corpus. At run-time, the input speech is selected from the source speaker's corpus. Our ARVC aims to modify the source speaker's voice to sound like that of the target speaker.

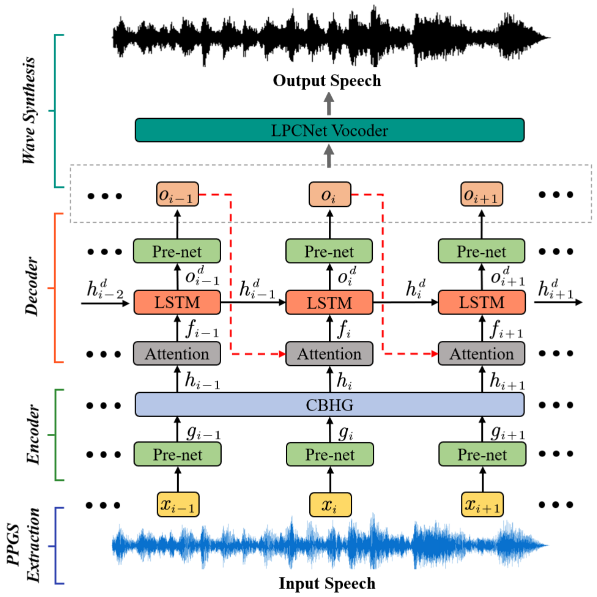


Figure 1: *Block diagram of the ARVC system architecture.*

2.1. Encoder

The input of the encoder is a sequence of PPGs, which is the frame-level linguistic information representation obtained from SI-ASR. We then apply a set of non-linear transformations to the inputs, calling pre-net [24]. The pre-net contains a bottleneck layer with dropout, which helps convergence and improves generalization. The outputs of the pre-net are fed into a CBHG module [24] to produce the final results of the encoder. The CBHG module is able to learn high-level context-dependent representations. It consists of a bank of 1-D convolutional filters, followed by highway networks and a bidirectional gated recurrent unit (GRU).

Specifically, we assume the inputs as $X = [x_1, x_2, ..., x_N]$, where $X \in \mathbb{R}^{N\times d_x}$. Here, N and d_x are used to represent the sequence length and feature dimensions of PPGs, respectively. The output sequence of the encoder $H = [h_1, h_2, ..., h_N]$ (where $H \in \mathbb{R}^{N\times d_h}$) is calculated as follows:

$$G = PreNet(X) \tag{1}$$

$$H = CBHG(G) \tag{2}$$

where $G = [g_1, g_2, ..., g_N]$, $G \in \mathbb{R}^{N\times d_g}$ is the outputs of the encoder's pre-net.

2.2. Decoder

The decoder is an auto-regressive model that predicts a sequence of acoustic features from the encoder outputs. It consists of an attention layer, an LSTM layer and a pre-net.

Specifically, we assume the acoustic features as $O = [o_1, o_2, ..., o_N]$, where $O \in \mathbb{R}^{N\times d_o}$. At every output step i, the attention layer produces the fusion representation $f_i \in \mathbb{R}^{1\times d}$, given the previous step acoustic features $o_{i-1} \in \mathbb{R}^{1\times d_o}$ and the encoder outputs $h_i \in \mathbb{R}^{1\times d_h}$. It is calculated as follows:

$$f_{cat} = [o_{i-1}W_o; h_iW_h] \tag{3}$$

$$P_f = tanh(f_{cat}W_f) \tag{4}$$

$$\alpha_{att} = softmax(P_f w_f) \tag{5}$$

$$f_i = \alpha_{att}^T f_{cat} \tag{6}$$

where $W_o \in \mathbb{R}^{d_o\times d}$ and $W_h \in \mathbb{R}^{d_h\times d}$ are trainable parameters. These matrices are used to equalize feature dimensions of all inputs to size d. Here, $f_{cat} \in \mathbb{R}^{2\times d}$ are concatenated representations. And $W_f \in \mathbb{R}^{d\times d}$ and $w_f \in \mathbb{R}^{d\times 1}$ are trainable parameters. $\alpha_{att} \in \mathbb{R}^{2\times 1}$ are attention vectors over two inputs.

Then, the LSTM layer produces outputs o_i^d, given the previous step hidden state h_{i-1}^d and the fusion representation f_i. Finally, the LSTM outputs o_i^d are passed into a pre-net, to predict acoustic features $o_i \in \mathbb{R}^{1\times d_o}$ for speech generation. These procedures are summarized as follows:

$$o_i^d = LSTM(f_i, h_{i-1}^d) \tag{7}$$

$$o_i = PreNet(o_i^d) \tag{8}$$

2.3. Waveform Synthesis

Vocoders influence the quality of converted speech. In this paper, we choose the LPCNet vocoder [23] for speech generation. LPCNet is a WaveRNN [25] variant that uses the neural networks to generate speech samples from Bark-Frequency Cepstral Coefficients (BFCCs) [26], pitch period and pitch correlation parameters. This has the advantages of better control over the outputs of the spectral shape since it depends directly on the linear predictive coding filter shape.

In this work, we use the code published by the Mozilla team [23] with some modifications. To better control high frequency features, we increase 18-dimensional BFCCs to 30-dimensional BFCCs. In the meantime, we utilize OpenBLAS to accelerate the LPCNet inference. Therefore, our acoustic features O contains 30-dimensional BFCCs, 1-dimensional pitch period and 1-dimensional pitch correlation. Totally, the feature dimension of the acoustic features is $d_o = 32$.

2.4. Model Training

We assume predicted acoustic features as $\hat{O} = [\hat{o}_1, \hat{o}_2, ..., \hat{o}_N]$ and the ground truth acoustic features as $O = [o_1, o_2, ..., o_N]$. We choose the L_2 loss function during training. The calculation formula is shown as follows:

$$L = \sum_{i=1}^{N} ||o_i - \hat{o}_i||^2 \tag{9}$$

In this paper, to reduce the mismatch between training and inference stages, we use the schedule sampling approach [27]. During training, at every output step i, we randomly decide whether we use the ground truth of the previous step o_{i-1} or the estimate $\hat{o}_{i-1}$, to generate the inputs of the decoder. At runtime, the predicted acoustic features of the previous step $\hat{o}_{i-1}$ is fed to produce the next step outputs.

3. Experimental Databases and Setup

In this section, we first present our experimental databases for voice conversion. Then, we illustrate implementation details of our ARVC. Finally, several baseline models are presented, and utilized to evaluate the performance of our proposed method.

3.1. Corpus Description

The CMU-ARCTIC American English database [28] is a popular benchmark database for voice conversion. It consists of 4 native American English speakers (2 male BDL and RMS, 2 female CLB and SLT), and each speaker has 1,134 sentences. Our voice conversion experiments are conducted on these four speakers. Intra-gender and inter-gender conversions are conducted between following pairs: RMS to BDL (M2M), CLB to SLT (F2F), SLT to BDL (F2M) and BDL to SLT (M2F). We use 1,000 sentences for training, and another 20 non-overlap utterances of each speaker are utilized for evaluation.

3.2. Implementation Details

To extract PPGs from the input speech, we use a time-delay neural network long short term memory (TDNN-LSTM) based acoustic model. This acoustic model is implemented using the Kaldi toolkit [29] and trained on our 20,000 hours corpus. The acoustic features used for training are 40-dimensional filter-bank features, computed with a 25ms window size and 10ms window shift. Outputs of the last LSTM layer are utilized as the frame-level lexical features, PPGs. Totally, 512-dimensional PPGs (where $d_x = 512$) are extracted for each input waveform.

In the ARVC encoder, the pre-net is a dense layer with a dropout of 0.5, and the output dimension is $d_g = 512$. The CBHG module has $K = 16$ sets of 1-D convolutional filters, where the k-th set contains 128 filters of width k ($k \in [1, K]$) with ReLU activation. These outputs are max pooled with stride of 1 and width of 2, followed by a convolution layer with width of 3 and 512 output channels. Then we add the CBHG's inputs to the convolutional outputs with a residual connection, followed by a dense layer with a 128 output dimension. The highway network consists of 4 layers of fully-connected layers with a 128 output dimension, followed by a bidirectional GRU (128 units for each GRU component). Finally, we concatenate the outputs of two GRU components together, thus generating 256-dimensional context-dependent lexical features ($d_h = 256$).

In the ARVC decoder, the attention layer fuses previous step acoustic features and encoder outputs, and the output dimension is $d = 256$. These fusion representations are fed into a 2-layer LSTM with 1024 cells. To improve the robustness to perturbations in the hidden state, LSTM layers are regularized using zoneout [30] with probability 0.1. To predict acoustic features for speech generation, we pass the outputs to the pre-net with a dropout of 0.5, and the output dimension is $d_o = 32$.

To optimize parameters, we use the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and the initial learning rate is 0.001, with the Noam decay scheme [31]. We train our models for at least 100k steps with a batch size of 32. Gradient clipping is also utilized for regularization with a norm set to 1.0.

3.3. Baseline Models

For comparison, we implement following state-of-the-art baseline approaches to comprehensively evaluate the performance of the proposed ARVC (***Proposed***):

System I (***S1***) [32]: It employs SI-ASR and Kullback-Leibler Divergence (KLD) based mapping approach to voice conversion without using the parallel training data. The acoustic difference between source and target speakers is equalized with SI-ASR. KLD is chosen as a distortion measure to find an appropriate mapping from each input source speaker's frame to that of the target speaker. Finally, the STRAIGHT vocoder [20] is used to generate the converted waveform.

System II (***S2***) [33]: It achieves top rank on naturalness and similarity in Voice Conversion Challenge 2018 [34]. Firstly, the acoustic features of the source speaker (including Mel-cepstral coefficients (MCCs), F0 and band aperiodicities (BAPs)) are converted toward the target speaker using an LSTM-based conversion model. Then, the waveform samples of the converted speech are synthesized by sending the converted acoustic features into the WaveNet vocoder built for the target speaker. We try our best to reproduce the work in [33]. However, compared with the original system in [33], there are still two major differences: (1) There is no manually correction for F0 extraction errors, nor removal of speech segments with irregular phonation. (2) Due to the limited training data for VC, Liu et al. [33] train a speaker-dependent WaveNet by adapting a pre-trained multi-speaker model for the target speaker. Differently, we have relatively enough data to train WaveNet. Therefore, we only train the WaveNet in ***S2*** using the target speech.

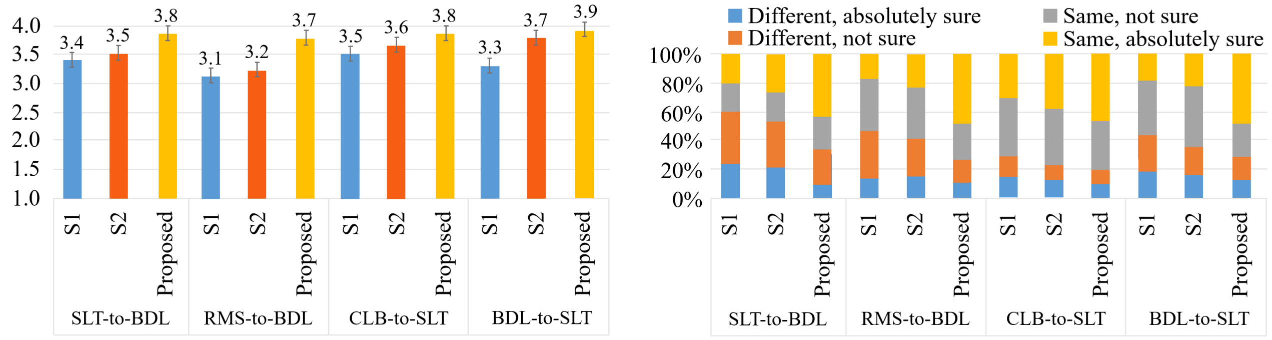
4. Results and Discussion

We compare our method with the state-of-the-art baseline approaches in terms of both objective and subjective measures. Speech samples from the following experiments are available online at https://zeroqiaoba.github.io/AR-voice-conversion.

4.1. Objective Evaluation

We conducted objective evaluation to assess the effectiveness of our ARVC. Mel-cepstral distortion (MCD) is employed as the objective measure to evaluate how close the converted speech is to the target speech. Concretely, MCD is the Euclidean distance between the Mel-cepstral coefficients (MCEPs) of the converted speech and the target speech. The MCEPs used for MCD calculation are 80-dimensional features, computed with a 25ms window size and 10ms window shift. Ideally, the lower MCD indicates the better performance of the voice conversion model. In this paper, MCD of one frame is calculated as follows:

$$MCD[dB] = \sqrt{\frac{1}{F}\sum_{f=1}^{F}\left(20log_{10}\frac{c_f}{c_f^{conv}}\right)^2} \tag{10}$$



(a) MOS test results with 95% confidence intervals to assess speech quality

(b) Same/Different paradigm to assess speaker similarity

Figure 2: *Subjective test results for our proposed method and two baseline models.*

Table 1: *Comparison of the Mel Cepstral Distortion (MCD) between the proposed ARVC and two baseline systems. Note: Bold front denotes the best performance.*

Conversion pairs	***S1***	***S2***	***Proposed***
SLT→BDL	9.16	8.34	**6.92**
RMS→BDL	10.48	9.82	**9.10**
CLB→SLT	10.79	10.50	**9.27**
BDL→SLT	10.37	10.23	**9.22**

where c_f and c_f^{conv} are the f^{th} coefficient of the target MCEPs and the converted MCEPs, respectively. And F is the feature dimension of the MCEPs.

The MCD results of objective evaluations are presented in Table 1. Experimental results show that our ARVC outperforms ***S1*** in all conversation pairs. The reason lies in that ***S1*** uses the STRAIGHT vocoder for speech generation. While ***S2*** adopts the LPCNet vocoder to reconstruct waveform from the converted acoustic features. These results verify the advantages of the LPCNet vocoder for speech generation. Meanwhile, our proposed method outperforms two baseline methods in all cases. By feeding previous step acoustic features to predict the next step outputs via the auto-regressive structure, our method generates more smooth trajectory of waveforms. Therefore, our ARVC achieves better performance than ***S1*** and ***S2***.

4.2. Subjective Evaluation

Following previous works [19, 35], the quality of the speech samples and their similarity to the target speaker are evaluated using the subjective evaluation. The Mean Opinion Score (MOS) tests are conducted to assess speech quality. In the MOS tests, listeners are asked to rate the converted speech on a 5-point scale, ranging from 1 (completely unnatural) and 5 (completely natural). Meanwhile, we conduct the Same/Different paradigm to assess speaker similarity. In this test, the listeners are asked to compare and select whether the converted samples are uttered by the same target speaker. In practice, 12 subjects with normal hearing participate in all tests. 20 utterances in the test set for each conversion pair are randomly selected and converted using our proposed method and two baseline methods. The listeners are asked to use headphones and samples are shown to them in a random order.

The results of subjective evaluations are presented in Figure 2. Figure 2(a) and Figure 2(b) show the MOS test results and similarity test results of the conversion performance on CMU-ARCTIC English database, respectively. Results showed in Figure 2 suggest that the performance of our proposed method significantly outperforms that of ***S1*** in all the conversion pairs. These results suggest that the LPCNet vocoder significantly outperforms that of the STRAIGHT vocoder in terms of speech quality and speaker similarity.

As shown in Figure 2, experimental results show that our proposed method achieves slightly better performance than ***S2*** in all cases. Meanwhile, from the additional listener's feedback, we observe that our proposed method suffers from less voicing error problems compared with ***S2***. Compared with ***S2***, our proposed method takes previous step acoustic features as the inputs to produce the next step outputs via the auto-regressive model. Therefore, our ARVC can generate more smooth trajectory and achieve better performance than ***S2***.

5. Conclusions

In this paper, a novel auto-regressive model is proposed for voice conversion, called auto-regressive voice conversion (ARVC). Compared with conventional PPGs based VC, ARVC takes previous step acoustic features as the inputs to produce the next step outputs via the auto-regressive structure. Then the LPCNet vocoder uses these predicted acoustic features to synthesize the speech waveform of the target speaker. To verify the effectiveness of our proposed method, we conduct experiments on the popular benchmark database, CMU-ARCTIC. Experimental results show that our proposed method outperforms two state-of-the-art baseline approaches in terms of speech quality and speaker similarity. Meanwhile, from the additional listener's feedback, we observe that our proposed method suffers from less voicing error problems.

6. Acknowledgements

This work is supported by the National Key Research & Development Plan of China (No.2017YFC0820602), the National Natural Science Foundation of China (NSFC) (No.61831022, No.61771472, No.61773379, No.61901473) and the Key Program of the Natural Science Foundation of Tianjin (Grant No. 18JCZDJC36300). The authors would like to thank Xinyong Zhou for constructive discussions and feedback.

7. References

[1] A. B. Kain, J.-P. Hosom, X. Niu, J. P. Van Santen, M. Fried-Oken, and J. Staehely, "Improving the intelligibility of dysarthric speech," *Speech communication*, vol. 49, no. 9, pp. 743–759, 2007.

[2] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 3277–3281.

[3] C. Deng, C. Yu, H. Lu, C. Weng, and D. Yu, "Pitchnet: Unsupervised singing voice conversion with pitch adversarial network," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 7749–7753.

[4] O. Turk and M. Schroder, "Evaluation of expressive speech synthesis with voice conversion and copy resynthesis techniques," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 965–973, 2010.

[5] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Transactions on speech and audio processing*, vol. 6, no. 2, pp. 131–142, 1998.

[6] R. Takashima, T. Takiguchi, and Y. Ariki, "Exemplar-based voice conversion using sparse representation in noisy environments," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 96, no. 10, pp. 1946–1953, 2013.

[7] S. Desai, A. W. Black, B. Yegnanarayana, and K. Prahallad, "Spectral mapping using artificial neural networks for voice conversion," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 954–964, 2010.

[8] F. Biadsy, R. J. Weiss, P. J. Moreno, D. Kanvesky, and Y. Jia, "Parrotron: An end-to-end speech-to-speech conversion model and its applications to hearing-impaired speech and speech separation," *Proceedings of the Interspeech*, pp. 4115–4119, 2019.

[9] R. Liu, X. Chen, and X. Wen, "Voice conversion with transformer network," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 7759–7759.

[10] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2014.

[11] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from non-parallel corpora using variational auto-encoder," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2016, pp. 1–6.

[12] W.-C. Huang, H.-T. Hwang, Y.-H. Peng, Y. Tsao, and H.-M. Wang, "Voice conversion based on cross-domain features using variational auto encoders," in *International Symposium on Chinese Spoken Language Processing*, 2018, pp. 51–55.

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

[14] T. Kaneko and H. Kameoka, "Cyclegan-vc: Non-parallel voice conversion using cycle-consistent adversarial networks," in *European Signal Processing Conference*, 2018, pp. 2100–2104.

[15] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "Cyclegan-vc2: Improved cyclegan-based non-parallel voice conversion," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 6820–6824.

[16] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "Stargan-vc: Non-parallel many-to-many voice conversion using star generative adversarial networks," in *IEEE Spoken Language Technology Workshop*, 2018, pp. 266–273.

[17] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "Stargan-vc2: Rethinking conditional methods for stargan-based voice conversion," 2019, pp. 679–683.

[18] L. Sun, K. Li, H. Wang, S. Kang, and H. Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2016, pp. 1–6.

[19] Y. Zhou, X. Tian, H. Xu, R. K. Das, and H. Li, "Cross-lingual voice conversion with bilingual phonetic posteriorgram and average modeling," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 6790–6794.

[20] H. Kawahara, I. Masuda-Katsuse, and A. De Cheveigne, "Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds," *Speech communication*, vol. 27, no. 3-4, pp. 187–207, 1999.

[21] 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.

[22] 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," in *The 9th ISCA Speech Synthesis Workshop*, 2016, p. 125.

[23] J.-M. Valin and J. Skoglund, "Lpcnet: Improving neural speech synthesis through linear prediction," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 5891–5895.

[24] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," *Proceedings of the Interspeech*, pp. 4006–4010, 2017.

[25] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. v. d. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," 2018, pp. 2415–2424.

[26] B. C. Moore, *An introduction to the psychology of hearing*. Brill, 2012.

[27] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Processings of Advances in Neural Information Processing Systems*, 2015, pp. 1171–1179.

[28] J. Kominek, A. W. Black, and V. Ver, "Cmu arctic databases for speech synthesis," 2003.

[29] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," IEEE Signal Processing Society, Tech. Rep., 2011.

[30] D. Krueger, T. Maharaj, J. Kramár, M. Pezeshki, N. Ballas, N. R. Ke, A. Goyal, Y. Bengio, A. Courville, and C. Pal, "Zoneout: Regularizing rnns by randomly preserving hidden activations," 2017.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[32] F.-L. Xie, F. K. Soong, and H. Li, "Voice conversion with si-dnn and kl divergence based mapping without parallel training data," *Speech Communication*, vol. 106, pp. 57–67, 2019.

[33] L.-J. Liu, Z.-H. Ling, Y. Jiang, M. Zhou, and L.-R. Dai, "Wavenet vocoder with limited training data for voice conversion." in *Proceedings of the Interspeech*, 2018, pp. 1983–1987.

[34] J. Lorenzo-Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, T. Kinnunen, and Z. Ling, "The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods," in *Odyssey 2018: The Speaker and Language Recognition Workshop*, 2018, pp. 195–202.

[35] X. Tian, E. S. Chng, and H. Li, "A speaker-dependent wavenet for voice conversion with non-parallel data," *Proceedings of the Interspeech*, pp. 201–205, 2019.